

Digital signature verification method

Patent number: DE19829643

Publication date: 1999-01-07

Inventor: OHTA KAZUO (JP); OKAMOTO TATSUAKI (JP)

Applicant: NIPPON TELEGRAPH & TELEPHONE (JP)

Classification:


- international: H04L9/32; H04L9/30

- european: H04L9/32S

Application number: DE1981029643 19980702

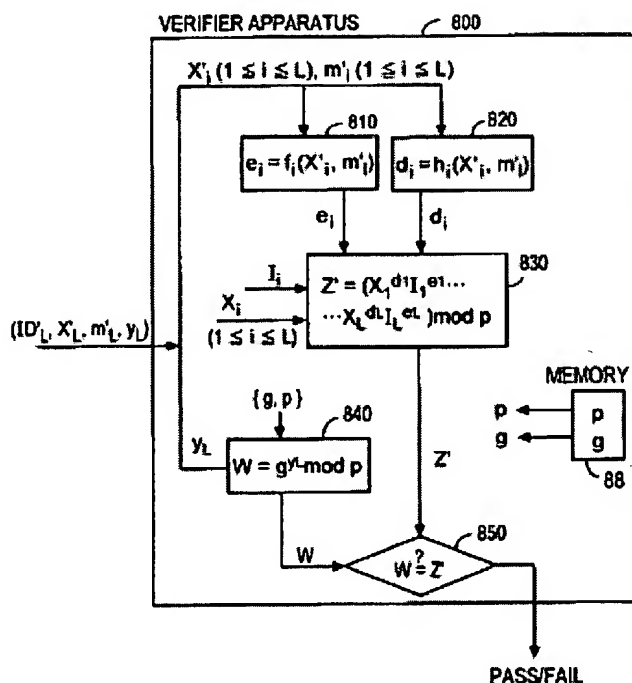
Priority number(s): JP19970179872 19970704; JP19970179873 19970704;
JP19970182724 19970708

Also published as:

 US6212637 (B1)

Abstract of DE19829643

The method involves generating a first random number as secret information by each signing person. Information are generated using a public parameter and the first random number. The information and two half-wave functions and identification information used by the signing person are published. A second random number is generated and second information are generated by inserting the public parameter and the second random number in a function. A signature is generated with a signature function which is generated with a parameter. Information which include identification information are sent to the next signing person in line. The last signing person sends the information to the verifier. The verifier calculates the first information from the public information with respect to the identification information and calculates the half-wave functions. The second information are calculated and checks if the signatures are valid by comparing two equations.



Data supplied from the esp@cenet database - Worldwide



①9 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENT- UND
MARKENAMT

①2 **Offenlegungsschrift**
①0 **DE 198 29 643 A 1**

⑤1 Int. Cl.⁶:
H 04 L 9/32
H 04 L 9/30

②1 Aktenzeichen: 198 29 643.6
②2 Anmeldetag: 2. 7. 98
④3 Offenlegungstag: 7. 1. 99

DE 198 29 643 A 1

③0 Unionspriorität:

9-179872 04. 07. 97 JP
9-179873 04. 07. 97 JP
9-182724 08. 07. 97 JP

⑦1 Anmelder:

Nippon Telegraph and Telephone Corp.,
Tokio/Tokyo, JP

⑦4 Vertreter:

Hoffmann, E., Dipl.-Ing., Pat.-Anw., 82166
Gräfelfing

⑦2 Erfinder:

Ohta, Kazuo, Tokio/Tokyo, JP; Okamoto, Tatsuaki,
Tokio/Tokyo, JP

⑤4 Verfahren und Vorrichtung zur Block-Verifikation mehrerer digitaler Signaturen und Speichermedium, auf dem das Verfahren gespeichert ist

- ⑤7 Bei Empfang einer Nachricht $(ID'_{i-1}, X'_{i-1}, m'_{i-1}, Y_{i-1})$ von einem Unterzeichner $(i-1)$ erzeugt ein Unterzeichner i eine Zufallszahl r_i , berechnet dann $x_i = e^r \bmod p$ unter Verwendung von öffentlichen Informationen p, q und g und setzt dann $X'_i = (X'_{i-1}, X_i)$, $m'_i = (m'_{i-1}, m_i)$, berechnet dann $e_i = f_i(X'_i, m'_i)$, $d_i = h_i(X'_i, m'_i)$ mit öffentlichen Einweg-Funktionen f_i und h_i , berechnet $y_i = (y_{i-1} + d_i r_i + e_i s_i) \bmod q$ unter Verwendung einer geheimen Zufallszahl s_i , setzt $ID'_i = (ID'_{i-1}, ID_i)$ und sendet Information $(ID'_i, X'_i, m'_i, Y'_i)$ an den nächsten Unterzeichner $(i+1)$. Ein Verifizierer berechnet e_i und d_i mit den Einweg-Funktionen f_i und h_i unter Verwendung von X'_i und m'_i , die in der empfangenen Information $(ID'_i, X'_i, m'_i, Y'_i)$ enthalten sind, und prüft, ob $e^x = x_i^{e_i} \dots x_i^{e_n} \bmod p$ und verifiziert dadurch Signaturen der Unterzeichner en-bloc.

DE 198 29 643 A 1

Die vorliegende Erfindung betrifft ein Verfahren und eine Vorrichtung, die es einem Verifizierer ermöglichen, eine Block-Verifikation (Verifikation en-bloc) von Einzelsignaturen, Mehrfachsignaturen oder Überlagerungssignaturen durchzuführen, die elektronisch von mehreren Unterzeichnern einem oder mehreren in elektronischer Form vorliegenden Dokumenten in einem System hinzugefügt wurden, das zur Entscheidungsfindung dient, indem das Dokument oder die Dokumente unter den Unterzeichnern umlaufen. Die Erfindung betrifft außerdem ein Speichermedium, auf dem das Verifikationsverfahren aufgezeichnet ist.

Ein typisches digitales Signaturschema verwendet das RSA-Verschlüsselungssystem (R.L. Rivest, et al., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, Band, 21, No. 2. Seiten 120-126 (1978)). Das RSA-Verschlüsselungssystem wird nachfolgend beschrieben.

Ein Unterzeichner A erzeugt einen Signaturschlüssel (d, N) und einen Verifikationsschlüssel (e, N), die die nachstehenden Bedingungen erfüllen

$$15 \quad N = P \times Q$$

$$e \times d \equiv 1 \pmod{L}, \text{ wobei } L = \text{LCM} \{(P-1), (Q-1)\}.$$

Dann veröffentlicht der Unterzeichner A den Verifikationsschlüssel, während er den Signaturschlüssel geheim hält. LCM (a, b) drückt dabei das kleinste gemeinsame Vielfache der ganzen Zahlen a und b aus, wobei angenommen wird, daß P und Q zwei verschiedene große Primzahlen sind. Weiterhin steht $a \equiv b \pmod{L}$ dafür, daß a-b ein Vielfaches von L ist.

Das RSA-Verschlüsselungssystem ist ein Verschlüsselungssystem, dessen Sicherheit auf der Schwierigkeit beruht, eine Zerlegung von N in Primzahlfaktoren durchzuführen, wenn N groß ist (dies wird später als das "Faktorzerlegungsproblem" bezeichnet). Es ist schwierig, die d-Komponente des geheimen Signaturschlüssels aus dem veröffentlichten Verifikationsschlüssel (e, N) zu errechnen.

Ein Verifizierer B hält den Verifikationsschlüssel (e, N) des Unterzeichners A in Verbindung mit dessen Identifikationsinformation (ID). Ein vertrauenswürdige Zentrum bzw. eine vertrauenswürdige Institution kann in machen Fällen solche Verifikationsschlüssel in der Form eines öffentlichen Informationsverzeichnisses halten.

30 Eine Signaturfunktion D und eine Verifikationsfunktion E sind wie folgt definiert:

$$D(m) = m^d \pmod{N}$$

$$E(y) = y^e \pmod{N}.$$

35 Man kann zeigen, daß die folgende Gleichung für eine ganze Zahl m erfüllt ist, für die gilt $0 \leq m < N$:

$$E(D(m)) = m,$$

40 wobei a mod N den Rest der Division a durch N darstellt.

Das digitale Signaturschema unter Verwendung des RSA-Verschlüsselungssystems ist wie nachfolgend beschrieben. Der Unterzeichner A erzeugt f(m) unter Verwendung einer Einweg-Funktion f aus einem Dokument m, fügt dann unter Verwendung der geheimen Signaturfunktion D eine Signatur $y = D(f(m))$ hinzu und sendet die Kombination (ID, m, y) seiner Identifikationsinformation (ID), des Dokuments m und der Signatur y als unterzeichnete Nachricht an den Verifizierer B.

Der Verifizierer B holt die Information über den Verifikationsschlüssel (e, N) des Unterzeichners A von dem öffentlichen Informationsverzeichnis unter Verwendung der Identifikationsinformation ID des Unterzeichners als Schlüssel, berechnet dann $E(y) = y^e \pmod{N}$ aus der y-Komponente der unterzeichneten Nachricht unter Verwendung des erhaltenen Verifikationsschlüssels (e, N) und prüft ob $E(y)$ mit f(m) übereinstimmt, welches mit Hilfe der Einweg-Funktion f von m abgeleitet wurde. Wenn $E(y) = f(m)$, urteilt der Verifizierer B, daß der Sender der echte Unterzeichner A ist und die unterzeichnete Nachricht (ID, m, y) nicht verfälscht wurde, da ausschließlich der wahre Unterzeichner A die Signaturfunktion $D(m) = m^d \pmod{N}$, d. h. die vorgenannte d-Komponente kennt.

Die hier erwähnte Einweg-Funktion f ist eine Funktion, mit der f(x) leicht aus x errechnet werden kann, während es schwierig ist, x aus f(x) zu ermitteln. Die Einweg-Funktion f kann unter Verwendung eines traditionellen schnellen Verschlüsselungssystems erstellt werden, beispielsweise eines DES-Verschlüsselungssystems (Data Encryption Standard, Federal Information Processing Standards Publication 46. 1977). Unter Verwendung schneller Komponenten wird die Zeit zur Errechnung der Funktion f so gut wie vernachlässigbar. Die nachfolgend erwähnte Einweg-Funktion ist eine solche, mit der ein Wert für ein x beliebiger Datenlänge errechnet werden kann.

Die ganze Zahl N zur Verwendung bei dem RSA-Verschlüsselungssystem umfaßt gewöhnlich eine Länge von 308 Dezimalstellen (1024 Bits) oder so. Die d-Komponente des Signaturschlüssels ist ebenfalls etwa 1024 Bits lang. Es ist im Stand der Technik bekannt, daß ein Quadrier- und-Multiplizier-Algorithmus zur Berechnung der Signaturfunktion d verwendet wird. Die Berechnung einer ganzen Zahl mit 308 Stellen (einschließlich einer Molulo-N-Rechnung) muß im Mittel 1536mal durchgeführt werden, was dem Unterzeichner A zur Signaturerzeugung einen schweren Rechenaufwand auferlegt.

65 Der Quadrier-und-Multiplizier-Algorithmus zur Berechnung von $x^a \pmod{N}$ ist wie nachfolgend beschrieben.

Schritt S1: $z = 1$

Schritt S2: Die folgenden Schritte S2-1 und S2-2 werden wiederholt, bis ein numerischer Index ausgehend von 0 la-1 wird (wobei angenommen wird, daß la die Anzahl von Bits von a darstellt).

- Schritt S2-1: $z' = z^2 \bmod N$
- Schritt S2-2: wenn $a_i = 1$, erneuere z mit $z = z'x \bmod N$ und kehre zum Schritt S2-1 zurück (a_i ist der Wert, 0 oder 1, des i -ten Bits von a);
- wenn $a_i = 0$, kehre zu Schritt S2-1 zurück, ohne z zu erneuern.
- Schritt S3: gib z aus. 5
- Die Quadrier- und Multiplizier-Algorithmus ist beispielsweise beschrieben in Douglas R. Stinson, "CRYPTOGRAPHY, Theory and Practice", CRC. Press p 127, 1995.
- Mit dem Ziel der Lösung des Problems der zunehmenden Rechenbelastung für den Unterzeichner zur Signaturerzeugung, sind interaktive Prüfungen vorgeschlagen worden, für die das Fiat-Shamir-Schema und das Schnorr-Schema typische Beispiele sind (vgl. A. Fiat und A. Shamir, "How to prove yourself: practical solutions to identification and signature problems", Advances in Cryptology-Crypto 86. Springer-Verlag, Seiten 186–194; C. F. Schnorr, "Efficient Identification and Signatures for smart Card", Advances in Cryptology-EUROCRYPT 79 Springer-Verlag Seiten 235–251; und M. Tompa und H. Woll, "Random Self-Reducibility and Zero Knowledge Interactive Proofs of Possession of Information", Proceedings of the 28th IEEE Symposium on the Foundation of Computer Science, Seiten 472–482 (1987)).
- Eine digitale Signatur mittels des Schnorr-Schemas wird nachfolgend beschrieben. 15
- Eine vertrauenswürdige Institution veröffentlicht zwei große Primzahlen p und q , die in einer solchen Beziehung miteinander stehen, daß q ein Maß von $p-1$ darstellt. Die Institution veröffentlicht außerdem eine ganze Zahl $g \in (\mathbb{Z}/p\mathbb{Z})^* = \{1, 2, \dots, p-1\}$ mit dem Grad q .
- Schritt S1: Der Unterzeichner A generiert eine Zufallszahl $s \in (\mathbb{Z}/q\mathbb{Z}) = \{0, 1, 2, \dots, q-1\}$, errechnet dann öffentliche Information I durch 20
- $$I = g^s \bmod p \quad (1)$$
- und veröffentlicht ein aus Identifikationsinformation (ID) und der Information I bestehendes Informationspaar.
- Der Unterzeichner A durchläuft die folgende Prozedur, um dem Verifizierer B zu beweisen, daß das Dokument oder die Nachricht m wahr bzw. echt ist. 25
- Schritt S2: Der Unterzeichner A generiert eine Zufallszahl $r \in (\mathbb{Z}/q\mathbb{Z})$ und berechnet
- $$X = g^r \bmod p \quad (2)$$
- Schritt S3: Der Unterzeichner A errechnet unter Verwendung der Einweg-Funktion f mittels der folgenden Gleichung eine ganze Zahl $e \in (\mathbb{Z}/q\mathbb{Z})$: 30
- $$e = f(X, m) \quad (3)$$
- Schritt S4: Der Unterzeichner A erzeugt die Signatur y durch 35
- $$y = r + er \bmod q \quad (4)$$
- und sendet $\{ID, m, X, y\}$ als unterzeichnete Nachricht an den Verifizierer B. 40
- Schritt S5: Der Verifizierer B errechnet unter Verwendung der Einweg-Funktion f die ganze Zahl $e \in (\mathbb{Z}/q\mathbb{Z})$ durch
- $$e = f(X, m) \quad (5)$$
- Schritt S6: Der Verifizierer B prüft, ob die nachfolgende Gleichung erfüllt ist. 45
- $$g^y \equiv XI^e \pmod{p} \quad (6)$$
- wobei I öffentliche Information entsprechend der Identifikationsinformation ID ist.
- Aus der Art der Erzeugung der Signatur y ergibt sich $g^y \equiv g^r (g^s)^e \equiv XI^e \pmod{p}$; wenn somit Gleichung (6) erfüllt ist, erkennt der Verifizierer B das Dokument m als von dem Unterzeichner A ordnungsgemäß unterzeichnet an. 50
- In den oben beschriebenen Schritten S2 bis S4 könnte die Signatur des Unterzeichners gefälscht werden, falls $\{ID, X, m, y\}$ als eine unterzeichnete Nachricht gesendet wird, wenn die ganze Zahl $e \in (\mathbb{Z}/q\mathbb{Z})$, für die $e = f(X, m)$ erfüllt ist, durch Berechnung von $X \in (\mathbb{Z}/p\mathbb{Z})^*$ herausgefunden werden könnte, welches Gleichung (6) erfüllen, nachdem die ganzen Zahlen $e \in (\mathbb{Z}/q\mathbb{Z})$ und $y \in (\mathbb{Z}/q\mathbb{Z})$ geeignet gewählt wurden. Da die Wahrscheinlichkeit, mit der die Verifikationsgleichung $e = f(m, X)$ erfüllt ist, jedoch $1/q$ ist, hängt der Komplexitätsgrad der mit der Fälschung der Signatur verbundenen Berechnung vom Wert q ab. In der folgenden Beschreibung wird die Anzahl von Bits der Primzahl p durch l_p dargestellt. 55
- Bei dem Schnorr-Schema beinhaltet der Signaturerzeugungsprozeß beim Sender eine Multiplikation (einschließlich Modulo- p -Rechnungen) von ganzen Zahlen mit l_p Bits mit einer durchschnittlichen Häufigkeit von $3/2l_p$, eine einzelne Multiplikation (einschließlich Modulo- q -Rechnungen) von ganzen Zahlen mit l_q Bits sowie eine einzelne Addition (einschließlich Modulo- q -Rechnungen) der ganzen Zahlen mit l_q Bits. 60
- Während bei dem Voranstehenden die unterzeichnete Nachricht $\{ID, X, m, y\}$ ist, ist es auch möglich, e anstelle von X zu verwenden, um $\{ID, e, m, y\}$ zu liefern. In diesem Fall erfolgt eine Prüfung daraufhin, ob die Relation $e = f(X, m)$ erfüllt ist, und zwar durch Berechnung von X durch $X = (g^y)(I^e)^{-1} \bmod p$. Wenn $l_e < l_X$, wird die Nachricht mit letzterem kürzer. 65
- Man betrachte nun den Fall, daß mehrere Unterzeichner verschiedener Dokumente auf der Basis der überlagerten Signatur oder Überlagerungssignatur unterzeichnen. Ein typisches Beispiel der Verwendung des Überlagerungssignatur-

schemas ist folgendes. Eine Zertifikationsinstitution (Autorität) CA garantiert beispielsweise die Korrespondenz zwischen der öffentlichen Identifikationsinformation ID und öffentlicher Information I des Unterzeichners durch eine digitale Signatur $T = D_{CA}(ID, I)$, die einem Dokument (ID, I) beigelegt ist, und sendet die Signatur T dem Unterzeichner. Der Unterzeichner generiert eine Signatur $D_{ID}(m, T)$, für das aus dem Dokument in und der Signatur T bestehende Paar, und zwar unter Verwendung der Geheiminformation entsprechend der öffentlichen Information I, und sendet die Signatur $D_{ID}(m, T)$ an den Verifizierer, wodurch dieser in die Lage versetzt wird, die Signatur $D_{ID}(m, T)$ des Unterzeichners und die Signatur T der Zertifikationsinstitution CA zu verifizieren.

Bei dem Überlagerungssignaturschema ist es wichtig, die von dem Unterzeichner zur Signaturerzeugung zu verarbeitende Informationsmenge gering zu halten, die von dem Verifizierer zur Signaturverifikation zu verarbeitende Informationsmenge niedrig zu halten und eine Zunahme der Signaturkomponenten zu verhindern.

Bei dem von dem RSA-Verschlüsselungssystem Gebrauch machenden digitalen Signaturschemata, unterzeichnen jeweilige Unterzeichner i Dokumente m_i in sequentieller Reihenfolge, um Information $D_L(m_L, \dots, D_2(f(m_2, D_1(f(m_1)))) \dots)$ zu schaffen, wodurch die Überlagerungssignaturfunktion implementiert wird. In diesem Fall stellt der zur Signaturerzeugung erforderliche große Rechenumfang ein Problem dar.

Bei direkter Anwendung des Schnorr-Schemas auf das Überlagerungssignaturschema wird es für möglich gehalten, ein Verfahren des Anfügens von Information $\{ID, X_i, y_i\}$ zu Dokumenten $(m_1, \dots, m_{i-1}, m_i)$ für jeden Unterzeichner i einzusetzen. Die X_i -Komponente ist lpl Bits lang und die y_i -Komponente lql Bits lang. Wenn L Unterzeichner unterzeichnen, wird schließlich eine Information mit $(lpl + lql) \times L$ Bits an eine Nachricht angefügt, d. h. die Identifikationsinformation IB von L Unterzeichnern und Dokumente (m_1, \dots, m_L) . Auch in diesem Fall führt die Vergrößerung der Signaturkomponente (X-Komponente, y-Komponente) zu einem Problem.

Als nächstes erfolgt eine Beschreibung des Mehrfachsignaturschemas, bei dem mehrere Unterzeichner ein Dokument in sequentieller Reihenfolge unterzeichnen. Mit dem digitalen Signaturschema unter Verwendung des RSA-Verschlüsselungssystems ist es möglich, das Mehrfachsignaturschema zu implementieren, wenn die mehreren Unterzeichner auf einer Signatur y einer Nachricht $\{ID, m, y\}$ nacheinander unterzeichnen (d. h. $D_L \dots D_1(f(m))$). Auch bei diesem Schema stellt sich das Problem eines hohen Rechenaufwands zur Signaturerzeugung.

Bei direkter Anwendung des Schnorr-Schemas auf das Mehrfachsignaturschema wird es für machbar gehalten, ein Verfahren einzusetzen, bei dem Information $\{ID, X, y\}$ einer Nachricht m für jeden Unterzeichner hinzugefügt wird. Die X-Komponente ist lpl Bits lang und die y-Komponente lql Bits lang. Wenn L Unterzeichner die Nachricht in sequentieller Reihenfolge unterzeichnen, wird schließlich Information mit $(lpl + lql) \times L$ Bits einer Nachricht hinzugefügt (der Identifikationsinformation von L Unterzeichnern und dem Dokument m). Auch in diesem Fall bewirkt eine Zunahme der Signaturkomponente (X-Komponente, y-Komponente) ein Problem.

Bezüglich des Mehrfachsignaturschemas ist ein solches vorgeschlagen worden, das die Verringerung sowohl der X- als auch der y-Komponente auf eine erlaubt, und zwar durch Akkumulieren der Werte der X- und der y-Komponenten für jeden Signaturerzeugungsprozeß (K. Ohta und T. Okamoto, "A Digital Multi-Signature Scheme Based on the Fiat-Shamir Scheme", Advances in Cryptology-ASIACRYPT'91, Springer-Verlag, Seiten 139-148). Da dieses Schema jedoch zwei Durchläufe einer Zirkulation der Nachricht zu den Unterzeichnern beinhaltet, erfordert die Mehrfachsignatur durch L Unterzeichner $(2L-1)$ Kommunikationsdurchläufe. Die Zunahme der Anzahl von Kommunikationen führt damit zu einem Problem.

Mit dem Mehrfachsignaturschema, das zwei Durchläufe der Zirkulation einer Nachricht zu Unterzeichnern beinhaltet, ist es unmöglich, das Überlagerungssignaturschema zu realisieren, bei dem die von den einzelnen Unterzeichnern jeweils zu unterzeichnenden Dokumente verschieden sind. Der Grund dafür besteht darin, daß, weil alle Dokumente, beispielsweise m_1 und m_2 , im ersten Zirkulationsdurchlauf bestimmt werden müssen, die Signatur der Dokumente (m_1, m_2) nicht nach Erzeugung der Signatur des Dokuments m_1 erzeugt werden kann.

Es ist ein Schema zur Modifizierung einer ElGamal-Signatur für die Mehrfachsignatur vorgeschlagen worden (Atsushi Shimbo, "Multi signature Schemes Based on the ElGamal Scheme", The 1994 Symposium on Cryptography and Information Security SCIS94-2C). Diese Literaturstelle beinhaltet jedoch nichts über die Überlagerungssignaturverwendung. Bei dem vorgeschlagenen modifizierten Schema ist es schwierig, die Schnorr-Signatur mit einem Zirkulationsdurchlauf zu realisieren, und die Sicherheit aller der vorgeschlagenen Schemata wurde nicht strikt bewertet (siehe "Conclusion" auf Seite 9 der Literaturstelle).

Bei einem das digitale Signaturschema verwendenden System tritt gelegentlich die Situation auf, wo mehrere Signaturen an einer Stelle zusammenkommen und verifiziert werden. Beispielsweise kommt elektronisches Geld zum Ausgabeinstitut zurück, wo seine Gültigkeit verifiziert wird. In einem solchen Fall erlaubt die Verwendung der interaktiven Prüfung eine wesentliche Verringerung des zur Signaturerzeugung erforderlichen Rechenaufwands. Jedoch kann die zu verarbeitende Rechenmenge zur Signaturverifikation manchmal zunehmen. Beispielsweise beinhaltet das Schnorr-Schema eine Multiplikation von ganzen Zahlen mit lpl Bits (einschließlich Modulo-p-Rechnungen) mit einer durchschnittlichen Häufigkeit von $3/2lql$, während bei dem RSA-Schema, da $e = 3$ ohne Beeinträchtigung der Sicherheit erreichbar ist, die Anzahl von Multiplikationen von ganzen Zahlen mit lnl Bits (einschließlich Modulo-N-Rechnungen) lediglich zwei beträgt.

Es soll nun eine Block-Verifikation mehrerer Signaturen bei den oben erwähnten digitalen Signaturschemata beschrieben werden.

Da das von dem RSA-Verschlüsselungssystem Gebrauch machende digitale Signaturschema auf das Problem des hohen Rechenaufwands zur Signaturerzeugung stößt und für jeden Unterzeichner einen anderen Modulowert N_i verwendet, wird es als unmöglich angesehen, N_i und N_j auf einmal zu verifizieren.

Wenn das Schnorr-Schema ohne Modifikationen verwendet wird, unterzeichnet der Unterzeichner i eine Nachricht m_i durch Hinzufügen der Information $\{ID_i, X_i, y_i\}$, wobei die X_i -Komponente lpl Bits lang ist und die y_i -Komponente lql Bits lang ist. In dem Fall, wo L Unterzeichner i jeweils ein anderes Dokument in unterzeichnen (mit $1 \leq i \leq L$) und L unabhängige Verifikationsgleichungen zum Verifizieren der L Signaturen verwendet werden, besteht der zu bearbeitende Rechenumfang für die Signaturverifikation in L Verifikationsdurchläufen.

Angesichts dessen ist ein Schema vorgeschlagen worden, welches die folgende eine Verifikationsgleichung verwendet, indem der Wert der y-Komponente akkumuliert wird:

$$g^{y'} = X_1 I_1^{e_1} \dots X_L I_L^{e_L} \pmod{p} \quad (7)$$

wobei

$$y' = \sum_{i=1}^L y_i \text{ und } e_i = f(X_i, m_i)$$

(siehe beispielsweise Ohta and Okamoto, "Multi-Signature Schemes Using Fiat-Shamir Scheme", Spring National Convention of the Institute of Electronics, Information and Communication Engineers of Jaoan (1989), A-277 (1989), und Harada and Tatebayashi, "An efficient method for computing a general monomial and its application", Technical Report of Institute of Electronics, Information and Communication Engineers of Japan ISEC91-40 (1991)).

Mit diesen Schemata kann jeder Unterzeichner die Signaturen anderer Unterzeichner fälschen, was zu Sicherheitsproblemen führt. Dieses Problem wird beispielsweise in Shimbo and Kawamura, "Consideration on computing vector addition chain and its application", Technical Report of the Institute of Electronics, Information and Communication Engineers of Japan ISEC91-59 (1991) erörtert.

In der obigen Literaturstelle sind die Signaturerzeugung, die Signaturverifikation und Angriffe darauf in der Situation beschrieben, in der mehrere Unterzeichner ein Dokument unterzeichnen. Selbst wenn jedoch jeder Unterzeichner ein anderes Dokument unterzeichnet, ermöglicht die Verwendung der oben erwähnten Verifikationsgleichung die direkte Anwendung des Angriffs auf die mehreren Signaturen, was zu einem entsprechenden Sicherheitsproblem führt.

Eine erste Aufgabe der vorliegenden Erfindung besteht darin, ein Signaturverfahren und eine Vorrichtung zu schaffen, die eine Block-Verifikation einer Überlagerungssignatur, einer Mehrfachsignatur oder von Einzelsignaturen gestatten, welche von mehreren Unterzeichnern an demselben oder verschiedenen Dokumenten angebracht sind, sowie ein Speichermedium zu schaffen, auf dem das Signaturverfahren gespeichert ist.

Eine zweite Aufgabe der vorliegenden Erfindung ist es, ein sicheres Überlagerungssignaturverfahren und eine dafür geeignete Vorrichtung zu schaffen, die eine Zunahme der Datenmenge für Signaturkomponenten in dem Fall verhindern, wo mehrere Unterzeichner jeweils ein anderes Dokument unterzeichnen und es erwünscht ist, die Reihenfolge der Unterzeichnung zu bestätigen, sowie ein Speichermedium zu schaffen, auf dem das Überlagerungssignaturverfahren gespeichert ist.

Eine dritte Aufgabe der vorliegenden Erfindung ist es, ein sicheres Mehrfachsignaturverfahren und eine dazu geeignete Vorrichtung zu schaffen, die die Realisierung einer Mehrfachsignatur mittels lediglich eines Umlaufs einer Nachricht zu mehreren Unterzeichnern gestatten und eine Zunahme der Datenmenge für Signaturkomponenten verhindern, sowie ein Aufzeichnungsmedium zu schaffen, auf dem das Mehrfachsignaturverfahren gespeichert ist.

Eine vierte Aufgabe der vorliegenden Erfindung ist es, ein sicheres Signaturverfahren und eine dazu geeignete Vorrichtung zu schaffen, die eine Block-Verifikation und damit effiziente Verifikation von Signaturen erlauben, wenn mehrere Unterzeichner jeweils ein anderes Dokument unterzeichnen, sowie ein Aufzeichnungsmedium zu schaffen, auf dem das Signaturverfahren gespeichert ist.

Ein Signaturverifikationsverfahren gemäß einem ersten Aspekt der vorliegenden Erfindung umfaßt die Schritte: Jeder Unterzeichner i:

(a) erzeugt eine erste Zufallszahl s_i als Geheiminformation, erzeugt dann Information $I_i = (s_i, \beta)$ mit einer Funktion G_2 unter Verwendung eines öffentlichen Parameters β und der ersten Zufallszahl s_i und veröffentlicht die Information I_i , zwei Einweg-Funktionen f_i und h_i und Identifikationsinformation ID_i , die von dem Unterzeichner i benutzt werden, als seine öffentliche Information $\{ID_i, I_i, f_i, h_i\}$;

(b) erzeugt eine zweite Zufallszahl r_i , erzeugt dann $X_i = \Phi(r_i, \beta)$ durch Einsetzen des Parameters β und der zweiten Zufallszahl r_i in eine Funktion Φ und setzt die Information X_i enthaltende Information auf X'_i ;

(c) erzeugt

$$e_i = f_i(X'_i, m'_i)$$

$$d_i = h_i(X'_i, m'_i)$$

mit den Einweg-Funktionen f_i und h_i unter Verwendung von Dokumentinformation m'_i , die ein zu unterzeichnendes Dokument m_i enthält, und der Information X'_i ; und

(d) erzeugt für Information, die e_i , d_i , s_i und r_i enthält, eine Signatur

$$y_i = Sg_i(e_i, d_i, s_i, r_i, y'_{i-1})$$

mit einer Signaturfunktion Sg_i , die unter Verwendung des Parameters β erzeugt wird, und gibt, wenn Information, die die Information ID_i enthält, als Identifikationsinformation ID'_i dargestellt wird, $\{ID'_i, X'_i, m'_i, y_i\}$ einzeln oder über die anderen Unterzeichner an einen Verifizierer als letzte Bestimmung aus, wobei im Fall des einzelnen Aussendens y'_{i-1} eine leere Menge ist und im Fall des Aussendens über die anderen Unterzeichner y'_{i-1} so eingestellt wird, daß gilt $y'_{i-1} = y_{i-1}$; und der Verifizierer:

(e) errechnet aus der öffentlichen Information $\{ID_i, I_i, f_i, h_i\}$ Information I_i , die der Identifikationsinformation ID_i entspricht, welche in ID'_i in der empfangenen Information $\{ID'_i, X'_i, m'_i, y_i\}$ enthalten ist, und die Einwegfunktionen

f_i und h_i , und berechnet e_i und d_i unter Verwendung der Einweg-Funktionen f_i und h_i und der empfangenen Informationen X'_i und m'_i ;
 (f) berechnet die Information X_i , die in der Information X'_i enthalten ist, und berechnet

$$Z' = V((X_i * d_i), (I_i * e_i) | i = 1, \dots, L)$$

mit einer Funktion V , die Berechnungen $(X_i * d_i)$ von d_i und X_i sowie $(I_i * e_i)$ von e_i und I_i enthält, für $i = 1, \dots, L$; und
 (g) berechnet $W = \Gamma(y_i * \beta)$ mit einer Funktion Γ , die eine Berechnung $(y_i * \beta)$ von y_i und β enthält, verifiziert dann die Gültigkeit der Signaturen durch Prüfung, ob $W = Z'$, und entscheidet, falls beide Werte gleich sind, daß die Signaturen alle gültig sind.

Gemäß einem zweiten Aspekt der vorliegenden Erfindung wird der Wert der y-Komponente, bei der es sich um eine der Hauptsignaturkomponenten handelt, für jeden Signaturerzeugungsprozeß akkumuliert, um eine Zunahme der Datenmenge der Gesamtsignaturkomponente zu unterdrücken, wodurch ein Überlagerungssignaturschema aufgestellt wird, das bei dem Fiat-Shamir-Schema und dem Schnorr-Schema anwendbar ist. Obwohl es bekannt ist, daß die Exponentialkomponente bei der Verifikationsverarbeitung lediglich die e-Komponente ist, die als eine Exponentialkomponente von I verwendet wird, führt die vorliegende Erfindung die d-Komponente als zweite Exponentialkomponente für die Potenzierung von X neu ein und erzeugt unter Berücksichtigung der Reihenfolge der Unterzeichner die e- und die d-Komponente, wodurch eine Zunahme der Anzahl von Kommunikationen verhindert wird und gleichzeitig Sicherheit geboten wird.

Gemäß einem dritten Aspekt der vorliegenden Erfindung wird der Wert der y-Komponente, bei der es sich um eine der Hauptsignaturkomponenten handelt, für jeden Signaturerzeugungsprozeß akkumuliert, um eine Zunahme der Datenmenge der Gesamtsignaturkomponente zu unterdrücken, wodurch ein Mehrfachsignaturschema erstellt wird, das auf das Fiat-Shamir-Schema und das Schnorr-Schema anwendbar ist. Während es bekannt ist, daß die Exponentialkomponente in der Verifikationsverarbeitung lediglich die e-Komponente ist, die als Exponentialkomponente von I verwendet wird, führt die vorliegende Erfindung neu die d-Komponente als eine zweite Exponentialkomponente für die Potenzierung von X ein, wodurch eine Zunahme der Anzahl von Kommunikationen verhindert wird, während gleichzeitig Sicherheit geboten wird.

Gemäß einem vierten Aspekt der vorliegenden Erfindung wird, während es bekannt ist, daß die Exponentialkomponente in der Verifikationsverarbeitung lediglich die e-Komponente ist, die als eine Exponentialkomponente von I verwendet wird, die d-Komponente neu als zweite Exponentialkomponente für die Potenzierung von X eingeführt, wodurch ein Signaturschema erstellt wird, welches eine Block-Signaturverifikation ermöglicht, die auf das Fiat-Shamir-Schema und das Schnorr-Schema anwendbar ist. Gleichzeitig wird Sicherheit geboten, selbst wenn der Wert der y-Komponente, bei der es sich um eine der Hauptsignaturkomponenten handelt zum Zeitpunkt der Signaturverifikation akkumuliert wird und lediglich eine Verifikationsgleichung verwendet wird.

Ausführungsbeispiele der Erfindung werden nachfolgend anhand der Zeichnungen näher erläutert. Es zeigen:

Fig. 1A ein Blockdiagramm, das den Aufbau eines Systems darstellt, bei dem das Überlagerungs- oder das Mehrfachsignaturschema und die Block-Signaturverifikation dafür gemäß der vorliegenden Erfindung anwendbar sind,

Fig. 1B ein Blockdiagramm, das den Aufbau eines Systems darstellt, bei dem das Einzelsignaturschema und die Block-Signaturverifikation dafür gemäß der vorliegenden Erfindung anwendbar sind,

Fig. 2 ein Blockdiagramm, das den funktionalen, mit einer Verarbeitung zur anfänglichen Informationseinstellung in Zusammenhang stehenden Aufbau einer Zentralvorrichtung 100 in Fig. 1A oder 1B darstellt,

Fig. 3 ein Blockdiagramm, das den funktionalen, mit einem Prozeß für die Systemeinschreibung in Zusammenhang stehenden Aufbau einer Unterzeichnervorrichtung in Fig. 1A zeigt,

Fig. 4 ein Diagramm, das eine Interaktionsfolge von Information mit überlagerten Signaturen zeigt,

Fig. 5 ein Blockdiagramm, das den funktionalen, mit einer Verarbeitung zur Signaturerzeugung in Zusammenhang stehenden Aufbau einer Unterzeichnervorrichtung in Fig. 1A zeigt,

Fig. 6 ein Blockdiagramm, das den funktionalen, mit der Verarbeitung zur Signaturverifikation in Zusammenhang stehenden Aufbau der Verifizierervorrichtung 800 in Fig. 1A zeigt,

Fig. 7 ein Diagramm zeigt, das eine Interaktionsfolge von Information mit Mehrfachsignaturen zeigt,

Fig. 8 ein Blockdiagramm, das den funktionalen, mit der Verarbeitung zur Signaturerzeugung in dem Mehrfachsignaturschema in Zusammenhang stehenden Aufbau der Unterzeichnervorrichtung in Fig. 1A zeigt,

Fig. 9 ein Blockdiagramm, das den funktionalen, mit der Verarbeitung zur Signaturverifikation in dem Mehrfachsignaturschema in Zusammenhang stehenden Aufbau der Verifizierervorrichtung 800 in Fig. 1A zeigt,

Fig. 10 ein Blockdiagramm, das den funktionalen, mit der Verarbeitung ihrer Systemeinschreibung in dem Einzelsignaturschema in Zusammenhang stehenden Aufbau einer Unterzeichnervorrichtung in Fig. 1B zeigt

Fig. 11 ein Diagramm, das eine Interaktionsfolge von Information in dem Einzelsignaturschema zeigt,

Fig. 12 ein Blockdiagramm, das den funktionalen, mit der Verarbeitung zur Signaturerzeugung in Zusammenhang stehenden Aufbau einer Unterzeichnervorrichtung in Fig. 1B zeigt,

Fig. 13 ein Blockdiagramm, das den funktionalen, mit der Verarbeitung zur Signaturverifikation in Zusammenhang stehenden Aufbau der Verifizierervorrichtung 800 in Fig. 1B zeigt,

Fig. 14 ein Blockdiagramm, das den funktionalen Aufbau der Zentralvorrichtung 100 in Fig. 1A zeigt der mit der Verarbeitung zur anfänglichen Informationseinstellung im Fall eines auf einer elliptischen Kurve beruhenden Verschlüsselungssystems in Zusammenhang steht,

Fig. 15 ein Blockdiagramm, das den funktionalen Block einer Unterzeichnervorrichtung in dem System von Fig. 1A zeigt der mit der Verarbeitung ihrer Systemeinschreibung unter Verwendung des auf einer elliptischen Kurve beruhenden Verschlüsselungssystems verbunden ist,

Fig. 16 ein Blockdiagramm, das den funktionalen Block einer Unterzeichnervorrichtung in dem System von Fig. 1A

zeigt, der mit der Verarbeitung zur Signaturerzeugung in dem Überlagerungssignaturschema unter Verwendung des auf einer elliptischen Kurve beruhenden Verschlüsselungssystems verbunden ist,

Fig. 17 ein Blockdiagramm, das den funktionalen Block der Verifizierervorrichtung 800 in dem System von Fig. 1A zeigt, der mit der Verarbeitung zur Signaturverifikation in dem Überlagerungssignaturschema unter Verwendung des auf einer elliptischen Kurve beruhenden Verschlüsselungssystems verbunden ist,

Fig. 18 ein Blockdiagramm, das den Aufbau einer Unterzeichnervorrichtung in einem System von Fig. 1A für das Mehrfachsignaturschema unter Verwendung einer elliptischen Kurve darstellt,

Fig. 19 ein Blockdiagramm, das den Aufbau der Verifizierervorrichtung 800 in dem System von Fig. 1A für das Mehrfachsignaturschema unter Verwendung der elliptischen Kurve zeigt,

Fig. 20 ein Blockdiagramm, das den Aufbau einer Unterzeichnervorrichtung in einem System gemäß Fig. 1B für das Einzelsignaturschema unter Verwendung der elliptischen Kurve zeigt,

Fig. 21 ein Blockdiagramm, das den Aufbau der Verifizierervorrichtung 800 in dem System von Fig. 1B für das Einzelsignaturschema unter Verwendung der elliptischen Kurve zeigt,

Fig. 22 eine Tabelle, die die grundlegenden Berechnungsgleichungen bei der vorliegenden Erfindung im Vergleich mit jenen in dem RSA-Schema und dem Schnorr-Schema zur Bewertung der vorliegenden Erfindung zeigt, und

Fig. 23 eine Tabelle, die den erforderlichen Rechenaufwand bei der vorliegenden Erfindung im Vergleich mit jenen beim RSA-Schema und beim Schnorr-Schema zeigt.

Das Block-Signatursystem gemäß der vorliegenden Erfindung umfaßt eine Zentralvorrichtung 100 (nachfolgend auch einfach als Zentrale bezeichnet), L Unterzeichnervorrichtungen (nachfolgend auch einfach als Unterzeichner bezeichnet) $30_1, 30_2, \dots, 30_L$, wobei L eine ganze Zahl gleich oder größer als 2 ist, und eine Verifizierervorrichtung 800 (nachfolgend auch einfach als Verifizierer bezeichnet). Die Unterzeichnervorrichtungen und die Verifizierervorrichtung sind jeweils über einen Kanal 400 garantierter Sicherheit mit der Zentralvorrichtung 100 verbunden. Die Unterzeichnervorrichtungen $30_1, 30_2, \dots, 30_L$ sind in einer Kette über Kanäle 500 nicht garantierter Sicherheit verbunden. Die L-te Unterzeichnervorrichtung 30_L ist mit der Verifizierervorrichtung 800 über einen Kanal 700 nicht garantierter Sicherheit verbunden. In diesem System bringt der Unterzeichner 30 seine Signatur an einem Dokument m_1 unter Verwendung einer Signaturfunktion Sg_1 an, sendet dann das unterzeichnete Dokument als $y_1 = Sg_1(m_1)$ an den nächsten Unterzeichner 30_2 , der seinerseits unter Verwendung einer Signaturfunktion Sg_2 seine Signatur an einem Dokument m_2 und der empfangenen Signaturinformation y_1 anbringt und sie als Signaturinformation $y_2 = Sg_2(m_1, y_1)$ an den Unterzeichner 30_3 sendet. Diese Verarbeitung wiederholt sich für jeden nachfolgenden Unterzeichner. Der letzte Unterzeichner 30_L bringt seine Signatur an einem Dokument m_L und der von ihm empfangenen Signaturinformation y_{L-1} unter Verwendung einer Signaturfunktion Sg_L an und sendet sie als Signaturinformation $y_L = Sg_L(m_L, y_{L-1})$ an die Verifizierervorrichtung 800. Eine solche Signaturverarbeitung wird als Überlagerungssignaturschema bezeichnet.

Wenn bei diesem Beispiel lediglich das Dokument m_1 des ersten Unterzeichners existiert, die Dokumente m_2, \dots, m_L der nachfolgenden Unterzeichner also nicht existieren, unterzeichnen die L Unterzeichner dasselbe Dokument m_1 nacheinander. Dies wird als Mehrfachsignaturschema bezeichnet. In jedem Fall verifiziert der Verifizierer 800 gemäß der vorliegenden Erfindung die empfangene Signaturinformation y_L en-bloc, d. h. alle auf einmal. Wenn alle Signaturen gültig sind, endet ihre Verifikation in einer Verarbeitungsrunde bzw. einem Verarbeitungsdurchlauf. Wenn jedoch irgend eine der Signaturen ungültig ist, wird eine Verarbeitung zur Ermittlung des nicht autorisierten Unterzeichners durchgeführt. Wenn beispielsweise die Anzahl involvierter Unterzeichner 2^M ist, werden diese in eine erste und eine zweite Halbgruppe je bestehend aus 2^{M-1} Unterzeichnern unterteilt, und die Signaturen von 2^{M-1} Unterzeichnern einer Gruppe werden auf einmal, d. h. als Block verifiziert. Wenn eine ungültige Signatur gefunden wird, werden 2^{M-2} Signaturen der ersten oder der zweiten Halbgruppe auf einmal verifiziert. Wenn keine ungültige Signatur gefunden wird, werden die Signaturen der verbleibenden 2^{M-1} oder 2^{M-2} Unterzeichner der ersten oder der zweiten Halbgruppe auf einmal verifiziert, wonach der gleiche Vorgang wiederholt wird. Auf diese Weise kann die unautorisierte Signatur in M+1 Durchläufen einer Verifikationsverarbeitung ermittelt werden.

Bei einem anderen System zur Ausführung der vorliegenden Erfindung ist gemäß Darstellung in Fig. 1B eine Zentralvorrichtung 100 mit L Unterzeichnervorrichtungen $30_1, 30_2, \dots, 30_L$ und einer Verifizierervorrichtung 800 über Kanäle 400 garantierter Sicherheit verbunden, wie dies auch bei Fig. 1A der Fall ist. Die Unterzeichnervorrichtungen $30_1, 30_2, \dots, 30_L$ sind hier jedoch je direkt mit der Verifizierervorrichtung 800 über einen jeweiligen Kanal 500 nicht garantierter Sicherheit verbunden. Bei diesem System bringt jeder Unterzeichner seine Signatur unter Verwendung einer Signaturfunktion an einem jeweiligen Dokument an (der i-te Unterzeichner an dem Dokument m unter Verwendung einer Signaturfunktion Sg_i) und sendet das unterzeichnete Dokument (im Fall des i-ten Unterzeichners als $y_i = Sg_i(m_i)$) an den Verifizierer 800, der die empfangene Information ($y_i = Sg_i(m_i)$ für $i = 1, \dots, L$) en-bloc verifiziert.

Nachfolgend werden der i-te, der (i-1)-te, der (i+1)-te, etc. Unterzeichner entsprechend der Unterzeichnervorrichtung $30_i, 30_{i-1}, 30_{i+1}$, etc. zur Vereinfachung als Unterzeichner i, Unterzeichner (i-1), Unterzeichner (i+1) etc. bezeichnet. Dabei gilt im Rahmen dieser Beschreibung $1 \leq i \leq L$, soweit nichts anderes ausgeführt ist.

Die Prinzipien der Verfahren zur Durchführung einer Block-Signaturverifikation von Signaturen mehrerer Unterzeichner bei den obigen beiden Systemen gemäß der vorliegenden Erfindung werden nachfolgend beschrieben.

Schritt S1: Die Zentrale 100 veröffentlicht (sendet an alle Unterzeichner und an den Verifizierer) öffentliche Parameterinformation enthaltend einen Parameter q für jeden Unterzeichner zur Erzeugung der Signaturfunktion Sg_i und einen Parameter $\beta = G_1(q)$, der unter Verwendung des Parameters q mit einer Funktion G_1 erzeugt wird.

Schritt S2: Jeder Unterzeichner i erzeugt eine erste Zufallszahl s_i als Geheiminformation und behält sie in einem Speicher. Weiterhin erzeugt der Unterzeichner i Information $I_i = G_2(s_i, \beta)$ mit einer Funktion G_2 unter Verwendung des öffentlichen Parameters β und der ersten Zufallszahl s_i und trägt die Information I_i , zwei Einweg-Funktionen f_i und h_i zur Verwendung durch den Unterzeichner i sowie seine Identifikationsinformation ID_i als öffentliche Unterzeichnerinformation $\{ID_i, I_i, f_i, h_i\}$ bei der Zentrale 100 ein.

Schritt S3: Der Unterzeichner i erzeugt eine zweite Zufallszahl r_i und setzt den Parameter β und die zweite Zufallszahl r_i in eine Funktion Φ ein, wodurch Information $X_i = \Phi(r_i, \beta)$ erzeugt wird. Information enthaltend die Information X_i

wird als X'_i eingestellt.

Schritt S4: Der Unterzeichner i verwendet Dokumentinformation m'_i , die ein zu unterzeichnendes Dokument m_i enthält, und die Information X'_i zur Erzeugung mittels der beiden Einweg-Funktionen f_i und h_i von

$$e_i = f_i(X'_i, m'_i) \quad (8)$$

$$d_i = h_i(X'_i, m'_i) \quad (9)$$

Schritt S5: Der Unterzeichner i erzeugt die folgende Signatur von Information enthaltend e_i , d_i , s_i , r_i und y'_{i-1} mit der Signaturfunktion Sg_i

$$y_i = Sg_i(e_i, d_i, s_i, r_i, y'_{i-1}), \quad (10)$$

setzt dann die Identifikationsinformation ID_i enthaltende Information als Identifikationsinformation ID'_i und sendet Information $\{ID'_i, X'_i, m'_i, y_i\}$ einzeln oder über die anderen Unterzeichner an den Verifizierer 800 als die letzte Bestimmung. Wenn einzeln an den Verifizierer gesendet wird, ist y'_{i-1} eine leere Menge, während wenn über die anderen Unterzeichner gesendet wird, y'_{i-1} so eingestellt wird, daß $y'_{i-1} = y_{i-1}$.

Schritt S6: Der Verifizierer 800 errechnet aus der öffentlichen Information $\{ID_i, I_i, f_i, h_i\}$ die Information I_i entsprechend der Identifikationsinformation ID_i , die in ID'_i in der empfangenen $\{ID'_i, X'_i, m'_i, y_i\}$ enthalten ist, sowie die beiden Einweg-Funktionen f_i und h_i , und berechnet e_i und d_i unter Verwendung der Einweg-Funktionen f_i und h_i und der empfangenen Informationen X'_i und m'_i gemäß Gleichungen (8) und (9). Weiterhin extrahiert der Verifizierer 800 X_i aus der empfangenen Information X'_i und führt dann eine Berechnung $d_i * X_i$ zwischen d_i und X_i sowie eine Berechnung $e_i * I_i$ zwischen e_i und I_i aus und berechnet den folgenden Wert anhand der Ergebnisse der obigen Berechnungen unter Verwendung einer Funktion V :

$$Z' = V((X_i * d_i), (I_i * e_i)_{i=1, \dots, L}) \quad (11).$$

Die mit dem Symbol $*$ bezeichnete Rechenvorschrift kann eine Potenzierung, Multiplikation oder ähnliches sein.

Schritt S7: Der Verifizierer 800 setzt weiterhin das Rechenergebnis $y_i * \beta$ zwischen y_i und β in eine Funktion Γ zur Errechnung von $W = \Gamma(y_i * \beta)$ ein und führt eine Signaturverifikation durch, indem geprüft wird, ob $W = Z'$. Wenn dies der Fall ist, entscheidet der Verifizierer, daß alle Signaturen gültig sind.

Im Fall der Durchführung einer Block-Signaturverifikation mittels des oben beschriebenen Verfahrens bei dem Überlagerungs- oder dem Mehrfachsignatursystem von Fig. 1A wird die Information $y'_{i-1} = y_{i-1}$ gesetzt und X'_i , m'_i und ID'_i wie folgt eingestellt:

$$X'_i = (X'_{i-1}, X_i) \quad (12)$$

$$m'_i = (m'_{i-1}, m_i) \quad (13)$$

$$ID'_i = (ID'_i, ID_{i-1}) \quad (14).$$

Der Unterzeichner i empfängt Information $\{ID'_{i-1}, X'_{i-1}, m'_{i-1}, y_{i-1}\}$ von dem vorhergehenden Unterzeichner $(i-1)$, führt dann die Schritte S3 bis S5 aus und sendet Information $\{ID'_i, X'_i, m'_i, y_i\}$ an den nächsten Unterzeichner $(i+1)$. Der letzte Unterzeichner führt die Schritte S3 bis S5 aus und sendet Information $\{ID'_L, X'_L, m'_L, y_L\}$ an den Verifizierer 800.

Wenn man bei dem Voranstehenden $m'_1 = m_1 = m$ und $m'_2 = m'_3 = \dots = m'_L =$ "leere Menge" einstellt, d. h. $m'_2 = m'_3 = \dots = m'_L = m$ einstellt, erhält man die oben genannte Mehrfachsignatur.

Im Fall der Block-Verifikation von Signaturen bei dem Einzelsignatursystem von Fig. 1B dadurch, daß man den Prozeduren der oben erwähnten Schritte S1 bis S7 folgt, werden $y'_{i-1} =$ leere Menge, $X'_i = X_i$, $m'_i = m_i$ und $ID'_i = ID_i$ eingestellt, und der Unterzeichner i sendet die Information $\{ID_i, X_i, m_i, y_i\}$, die mittels der Schritte S3, S4 und S5 erzeugt wird, direkt an den Verifizierer 800.

Wie oben in Verbindung mit Schritt S4 beschrieben, erzeugt gemäß der vorliegenden Erfindung jeder Unterzeichner zwei Informationen oder Komponenten e_i und d_i unter Verwendung der beiden Einweg-Funktionen f_i und h_i und erzeugt die Information y_i , die diese Komponenten enthält, während die Signaturverifikation unter Berücksichtigung dieser beiden Informationen e_i und d_i durchgeführt wird. Somit kann die Verifikation auf der Basis einer Informationsumlaufunde nach Unterzeichnung durch die Unterzeichner 1 bis L durchgeführt werden. Weiterhin ist die Sicherheit garantiert. Im Gegensatz dazu, führt, da das oben beschriebene Signaturverifikationsverfahren von Schnorr die Signatur y unter Verwendung der mittels einer Einweg-Funktion f errechneten e -Komponente und der Zufallszahlen r und s gemäß den Gleichungen (3) und (4) ableitet, die direkte Anwendung dieses Verfahrens auf das Überlagerungssignaturschema zu einer Vergrößerung der Informationsmenge $\{ID_i, X_i, y_i\}$, die von jedem Unterzeichner an den nächsten gesandt wird, wodurch unvermeidlich der von dem Verifizierer zur Signaturverifikation zu bewältigende Rechenaufwand erhöht wird.

Als nächstes folgt eine Beschreibung eines konkreten Verfahrens zur Ausführung des oben beschriebenen grundsätzlichen Block-Signaturverifikationsschemas bei den Systemen der Fig. 1A und 1B, sowie von Beispielen der einzelnen Unterzeichnungsvorrichtungen und der Verifizierervorrichtung zur Verwendung in dem Schema.

Ausführungsbeispiel 1

Dieses Ausführungsbeispiel betrifft die Anwendung der Überlagerungssignatur und der Block-Signaturverifikation entsprechend den Prinzipien der vorliegenden Erfindung auf die Schnorr-Schemata in dem System von Fig. 1A. Die Idee

der Verwendung einer zweiten Exponentialkomponente, die hier erwähnt wird, ist auch in weitem Umfang anwendbar auf die Fiat-Shamir-Schemata und auf digitale Signaturschemata, die interaktive Prüfungen einschließlich der Fiat-Shamir-Schemata verwenden. Beispiele der interaktiven Prüfungen einschließlich Fiat-Shamir-Schemata oder von ähnlichen sind in der oben erwähnten Literaturstelle M. Tompa und H. Woll beschrieben.

Zum Zeitpunkt seiner Einschreibung im System erzeugt jede Unterzeichner i Geheiminformation s_i und öffentliche Information und trägt öffentliche Information (ID, I) in einem öffentlichen Informationsverwaltungsverzeichnis der Zentrale 100 ein. Die Zentrale 100 sendet die öffentliche Information nach Bedarf an die Unterzeichner (Unterzeichnervorrichtungen $30_1, \dots, 30_L$) und den Verifizierer 800.

Als erstes soll die anfängliche Informationseinstellverarbeitung durch die Zentrale 100 zum Zeitpunkt des Systemstarts beschrieben werden (siehe Fig. 2). Diese Verarbeitung ist dazu vorgesehen, einen einzigartigen oder eindeutigen Wert $\{p, q, g\}$ des Systems zu veröffentlichen.

(1-A) Anfängliche Informationseinstellverarbeitung (durch die Zentrale beim Systemstart)

Schritt S1: Die Zentrale 100 erzeugt mittels eines Primzahlgenerators 110 eine Primzahl p und mittels eines Dividierers 120 eine Primzahl q , die ein Maß von $p-1$ ist.

Schritt S2: Die Zentrale 100 erzeugt ein Grundelement α von $(\mathbb{Z}/p\mathbb{Z})^*$ mittels eines Grundelementengenerators 130 und eine Ganzzahl g eines Grads oder einer Ordnung q unter Verwendung eines Modulo-Exponentiators 140 gemäß der folgenden Gleichung:

$$g = \alpha^{(p-1)/q} \bmod p \quad (15)$$

Die rechte Seite der Gleichung (15) stellt die zuvor erwähnte Funktion G_1 dar, während g auf der linken Seite β entspricht.

Schritt S3: Die öffentliche Information $\{p, q, g\}$ wird an die Unterzeichnervorrichtungen $30_1, \dots, 30_L$ und an die Verifizierervorrichtung 800 über die sicheren Kanäle 400 gesandt.

(1-B) Verarbeitung beim Unterzeichner i für dessen Einschreibung im System

Als nächstes erfolgt eine Beschreibung der Verarbeitung, die der Unterzeichner i durchführt, wenn er sich im System einschreibt (siehe Fig. 3, die die Unterzeichnervorrichtung 30_i entsprechend dem Unterzeichner i zeigt). Im Speicher 33 jeder Unterzeichnervorrichtung 30_i ist die von der Zentrale 100 empfangene öffentliche Information $\{p, q, g\}$ gespeichert.

Schritt S4: Der Unterzeichner i erzeugt die Zufallszahl s_i mittels eines Zufallsgenerators 31 und gibt sie in einen Modulo-Exponentiator 32 zusammen mit den öffentlichen Informationen g und p ein. Daraufhin wird die öffentliche Information I_i gemäß nachstehender Gleichung (16) berechnet:

$$I_i = g^{s_i} \bmod p \quad (16).$$

Die rechte Seite von Gleichung (16) stellt die oben erwähnte Funktion G_2 dar.

Schritt S5: Der Unterzeichner i sendet die Identifikationsinformation ID_i , die öffentliche Information I_i und die Einweg-Funktionen f_i und h_i über den sichere Kanal 400 an die Zentrale 100, um sie als öffentliche Information $\{ID_i, I_i, f_i, h_i\}$ registrieren zu lassen. Der Unterzeichner i hält die Zufallszahl s_i als geheime Information in dem Speicher 33.

Die anderen Unterzeichner (1 bis $(i-1)$ und $(i+1)$ bis L) führen dieselbe Verarbeitung aus, wenn sie Teilnehmer des Systems werden. Die Zentrale 100 liefert die öffentliche Information $\{ID_i, I_i, f_i, h_i\}$ auf irgendeine Weise, beispielsweise in der Form eines öffentlichen Verzeichnisses oder einer öffentlichen Datei, an den Verifizierer 800.

Bei der folgenden Beschreibung wird die unterzeichnete Version des Dokuments m'_i , die von dem Unterzeichner i geliefert wird, durch $\{ID'_i, X_i, m'_i, y_i\}$ identifiziert. Es erfolgt nun eine Beschreibung des Falls, wo der Unterzeichner $(i-1)$ die zu unterzeichnende Nachricht sendet und der Unterzeichner i seine Signatur an der Nachricht anbringt und die unterzeichnete Nachricht an den nächsten Unterzeichner $(i+1)$ sendet. Wenn L Unterzeichner eine Überlagerungssignatur erzeugen, braucht lediglich i schrittweise um eins von 1 auf L erhöht zu werden und die folgende Prozedur wiederholt zu werden. In diesem Fall wird der Unterzeichner $(L+1)$ als der Verifizierer betrachtet, $ID'_0 =$ leere Menge, $X'_0 =$ leere Menge und $y_0 = 0$.

(1-C) Verarbeitung beim Unterzeichner i zur Signaturerzeugung

Fig. 4 zeigt eine Interaktionsfolge einer Nachricht und Fig. 5 den funktionalen Aufbau der Unterzeichnervorrichtung 30_i . Wenn der Unterzeichner i eine Nachricht $\{ID'_{i-1}, X'_{i-1}, m'_{i-1}, y_{i-1}\}$ von dem Unterzeichner $(i-1)$ erhält, führt er die nachstehend beschriebene Signaturerzeugungsverarbeitung aus.

Schritt S6: Der Unterzeichner i erzeugt die Zufallszahl r_i mittels eines Zufallsgenerators 310 und gibt sie zusammen mit den öffentlichen Informationen $\{p, g\}$, die im Speicher 33 gespeichert sind, in einen Modulo-Exponentiator 320 ein, der die Funktion Φ berechnet, und in welchem X_i gemäß Gleichung (17) berechnet wird:

$$X_i = \Phi(r_i, g) = g^{r_i} \bmod p \quad (17).$$

Schritt S7: Der Unterzeichner i verwendet einen f_i -Funktionsrechner 330 und einen h_i -Funktionsrechner 340 zur Errechnung zweier Informationen e_i bzw. d_i gemäß

$$e_i = f_i(X'_i, m'_i) \quad (18)$$

$$d_i = h_i(X'_i, m'_i) \quad (19).$$

- 5 In diesem Fall gilt $X'_i = (X'_{i-1}, X_i)$ oder $m'_i = (m'_{i-1}, m_i)$, wobei m_i das vom dem Unterzeichner i zu unterzeichnende Dokument ist.

Schritt S8: Der Unterzeichner i gibt diese Informationen e_i , d_i und die Zufallszahl r_i in einen Modulo-Exponentiator **350** und dann in einen Modulo-Addierer **360** zusammen mit der öffentlichen Information q und der geheimen Information s_i ein, wodurch gemäß Gleichung (20) die Signatur erzeugt wird:

$$10 \quad y_i = (y_{i-1} + d_i r_i + e_i s_i) \bmod q \quad (20)$$

Die rechte Seite der Gleichung (20) stellt die Signaturfunktion Sg_i in Gleichung (10) dar.

- 15 Schritt S9: Der Unterzeichner i setzt $ID'_i = (ID'_{i-1}, ID_i)$ und sendet Information $\{ID'_i, X'_i, m'_i, y_i\}$ an den nächsten Unterzeichner $(i+1)$.

(1-D) Verarbeitung beim Verifizierer zur Signaturverifikation

- 20 Fig. 6 zeigt den funktionalen Aufbau der Verifizierervorrichtung **800**. Wenn der Verifizierer die Nachricht $\{ID'_L, X'_L, m'_L, y_L\}$ von dem Unterzeichner L ($i = L$) empfängt, verifiziert er die Gültigkeit der einzelnen Signaturen mittels der nachfolgend beschriebenen Verarbeitung.

- Schritt S10: Die Einweg-Funktionen f_i und h_i , die in der öffentlichen Information $\{ID_i, I_i, f_i, h_i\}$ enthalten sind, welche von der Zentrale **100** geliefert wird, werden in Einweg-Funktionsrechnern **810** bzw. **820** eingestellt. Die ersten i Komponenten der Information X'_L werden zur Bildung von X'_i verwendet, und die ersten i Komponenten der Information m'_L werden zur Bildung von m'_i verwendet. Die Informationen X'_i und m'_i , die auf diese Weise erhalten werden, werden in dem f_i -Funktionsrechner **810** und dem h_i -Funktionsrechner **820** eingestellt, mit denen die Komponenten e_i bzw. d_i durch die nachstehenden Gleichungen errechnet werden:

$$30 \quad e_i = f_i(X'_i, m'_i) = f_i(X_1, X_2, \dots, X_i, \{m_1, m_2, \dots, m_i\}) \quad (21)$$

$$d_i = h_i(X'_i, m'_i) = h_i(X_1, X_2, \dots, X_i, \{m_1, m_2, \dots, m_i\}) \quad (22).$$

- Schritt S12: Die Information I_i wird aus der öffentlichen Information $\{ID_i, T_i, f_i, h_i\}$ abgeleitet, die von der Zentrale **100** geliefert wird, und außerdem wird die Information X_i aus der Information X'_L abgeleitet. Diese Informationen I_i und X_i werden zusammen mit den Komponenten e_i und d_i sowie der öffentlichen Information p in einen Multikomponenten-Modulo-Exponentiator **830** eingegeben, in welchem gemäß nachstehender Gleichung Z' errechnet wird:

$$Z' = X'_1 I_1 e_1 \dots X'_L I_L e_L \bmod p \quad (23).$$

- 40 Die rechte Seite der Gleichung (23) entspricht der Funktion $V((X'_i * d_i), (I_i * e_i))$, auf die zuvor in Verbindung mit den Prinzipien der vorliegenden Erfindung verwiesen wurde.

Schritt S12: Die Information y_L und die öffentlichen Informationen p und g , die in dem Speicher **88** gespeichert sind, werden in einen Modulo-Exponentiator **840** eingegeben, um mittels der nachstehenden Gleichung (24) W zu errechnen:

$$45 \quad W = g^{y_L} L \bmod p \quad (24).$$

Schritt S13: Z' und W werden in einen Komparator **850** eingegeben, wo sie miteinander verglichen werden um sicherzustellen, daß

$$50 \quad W = Z' \quad (25).$$

Wenn sie einander gleich sind, wird davon ausgegangen, daß die Dokumente (m_1, \dots, m_L) ordnungsgemäß jeweils durch die L autorisierten Unterzeichner unterzeichnet wurden.

55 (1-E) Verbesserter Quadrier-und-Multiplizier-Algorithmus für Mehrfachkomponenten

Nachstehend erfolgt eine Beschreibung eines verbesserten Quadrier-und-Multiplizier-Algorithmus für die Berechnung von Gleichung (23) durch den Multikomponenten-Modulo-Exponentiator **830**, wie etwa eine Multikomponenten-Modulo-Potenzierung wie sie durch $x^a y^b \bmod N$ ausgedrückt ist.

- 60 Schritt 1: $z = 1$

Schritt 2: Die folgende Verarbeitung wird für den Index $i = 0, 1, \dots, |a|-1$ ausgeführt (wobei $|a|$ die Anzahl von Bits von a repräsentiert).

$$65 \quad \text{Schritt 2-1: } z = z^2 \bmod N \quad (26)$$

Schritt 2-2:

$$\text{wenn } (a_i, b_i) = (1, 0), z = zx \bmod N \quad (27)$$

wenn $(a_i, b_i) = (0, 1)$, $z = zy \bmod N$ (28)

wenn $(a_i, b_i) = (1, 1)$, $z = z(xy) \bmod N$ (29)

5

wobei a_i der Wert, 0 oder 1, eines i -ten Bits ist und dasselbe für b_i gilt.

Schritt 3: z wird ausgegeben.

Durch Verwendung des obigen Algorithmus mit $x = X_i$, $a = d_i$, $y = I_i$, $b = e_i$ und $N = p$, ist es möglich $Z_1 Z_2 \bmod p$ zu erhalten (wobei $Z_1 = X_1^d I_1^e \bmod p$).

Berücksichtigt man den Weg zur Erzeugung von y_L , gilt

10

$$g^{y_L} = g^{y_{L-1}} (g^{r_{L-1}})^{d_L} (g^{a_L})^{e_L} = g^{y_{L-1}} X_{L-1}^d I_{L-1}^e = \dots = X_1^d I_1^e \dots X_L^d I_L^e \bmod p \quad (30)$$

Wenn somit die Dokumente $\{m_1, \dots, m_L\}$ den obigen Test durch den Komparator 850 bestehen, akzeptiert der Verifizierer 800 die Dokumente als von den L autorisierten oder gültigen Unterzeichnern ordnungsgemäß unterzeichnet.

15

Ein Verfahren zum Implementieren des Multikomponenten-Quadrier- und Multiplizier-Algorithmus mit höherer Effizienz ist beispielsweise beschrieben in D. E. Knuth, "The Art of Computer Programming, Vol. 2, Seminumerical Algorithms", Addison-Wesley Publishing, (1981), P. 456, Exercises 27 und 35.

Gemäß dem in der voranstehenden Literaturstelle vorgeschlagenen Verfahren wird, wenn s als Speichereinheit zur Speicherung von Ergebnissen einer Vorberechnung in eine Tabelle eingestellt wird ($s = 2$ bei dem oben beschriebenen Multikomponenten-Quadrier- und Multiplizier-Algorithmus), die Anzahl von Multiplikationen (einschließlich Modulo- p -Rechnungen) wie folgt:

20

$$(2^s - s - 1) [(2L+1)/s] + [(2L+1)/s] |q| - 1 + |q| - 1$$

25

dabei ist unter $[b/a]$ die kleinste ganze Zahl zu verstehen, die größer als b/a ist.

Es ist auch möglich, das System so auszugestalten, daß jeder Unterzeichner i vor Erzeugung seiner Signatur prüft, ob die von ihm empfangene Nachricht $\{ID_{i-1}, X_{i-1}, m'_{i-1}, y_{i-1}\}$ von den vorhergehenden Unterzeichnern 1 bis $(i-1)$ ordnungsgemäß unterzeichnet wurde. Wenn dies der Fall ist, fügt er seine Signatur der verifizierten Nachricht bei. In diesem Fall liefert die Zentrale 100 die öffentliche Information (ID_i, I_i, f_i, h_i) für $i = 1, \dots, (i-1)$ vorab an den Unterzeichner i , und die Verifikation kann in gleicher Weise wie in den Schritten S10 bis S13 bei dem Verifizierer 800 durchgeführt werden. In diesem Fall wird L in den Schritten S10 bis S13 durch $(i-1)$ ersetzt.

30

Die Unterzeichnervorrichtungen und die Verifizierervorrichtungen führen die oben beschriebene Verarbeitung gewöhnlich mit Hilfe von Computern aus.

Wie zuvor erwähnt, ist die vorliegende Erfindung anwendbar nicht nur auf die Schnorr-Schemata, sondern auch auf die Fiat-Shamir-Schemata und auf digitale Signatur-Schemata, die interaktive Prüfungen einschließlich der Fiat-Shamir-Schemata verwenden. Demgemäß kann das Verfahren der vorliegenden Erfindung allgemein wie folgt zusammengefaßt werden:

35

Die Systemparameter, die veröffentlicht werden, sind p zur Angabe der Anzahl von Elementen der Gruppe, ein Element g der Gruppe, mit dem eine Gruppenberechnung beginnt, und eine positive ganze Zahl q derart, daß wenn das Element g q -mal berechnet wird, die Rechnung zu dem Element g zurückkehrt.

40

Der Unterzeichner i erzeugt die Zufallszahl s_i mittels des Zufallsgenerators im Moment seiner Einschreibung in das System und gibt die Zufallszahl s_i und die öffentlichen Informationen g und p in einen Gruppenrechner ein, in welchem das Element g s_i -mal gerechnet wird, um die öffentliche Information I_i zu berechnen; und veröffentlicht die öffentliche Information I_i und die Einweg-Funktionen f_i und h_i zusammen mit der Identifikationsinformation ID_i , behält jedoch die Zufallszahl s_i als geheime Information.

45

Bei der Signaturerzeugungsverarbeitung nach Empfang der unterzeichneten Nachricht $\{ID_{i-1}, X_{i-1}, m'_{i-1}, y_{i-1}\}$ vom Unterzeichner $(i-1)$ auf der Basis der Nachricht m_{i-1} , macht der Unterzeichner i folgendes:

er erzeugt die Zufallszahl r_i unter Verwendung des Zufallsgenerators, gibt sie dann zusammen mit den öffentlichen Informationen p und g in den Gruppenrechner ein, um das Element g r_i -mal zum Erhalt der Information X_i zu rechnen, und setzt $X'_i = (X_{i-1}, X_i)$ und $m'_i = (m'_{i-1}, m_i)$;

50

er berechnet die Komponenten e_i und d_i durch

$$e_i = f_i(X'_i, m'_i)$$

55

$$d_i = h_i(X'_i, m'_i)$$

unter Verwendung des f_i -Funktionsrechners und des h_i -Funktionsrechners; und

er gibt die Information e_i , d_i und r_i zusammen mit der öffentlichen Information q und der geheimen Information s_i in einen Exponentialkomponenten-Multiplizierer und einen Exponentialkomponenten-Addierer ein, worin y_i mit der Signaturfunktion Sg_i errechnet wird, und zwar durch

60

$$y_i = (y_{i-1} + d_i r_i + e_i s_i) \bmod q,$$

und er setzt dann $ID'_i = (ID_{i-1}, ID_i)$ und sendet die Nachricht $\{ID'_i, X'_i, m'_i, y_i\}$ an den nächsten Unterzeichner $(i+1)$.

65

Andererseits bildet der Verifizierer, wenn er die Nachricht $\{ID'_L, X'_L, m'_L, y_L\}$ von dem Unterzeichner L empfängt, X'_i aus den ersten i Komponenten der Information X'_L und m'_i aus den ersten i Komponenten der Information m'_L und gibt diese Informationen X'_i und m'_i dann in den f_i -Funktionsrechner und den h_i -Funktionsrechner ein, worin

$$e_i = f_i(X'_i, m'_i)$$

$$d_i = h_i(X'_i, m'_i)$$

- 5 berechnet werden, um für jedes i ($1 \leq i \leq L$) die Komponenten e_i und d_i zu erhalten. Der Verifizierer leitet dann die entsprechende öffentliche Information I_i aus der ID_i -Komponente in der Information ID'_L und die Information X_i aus der X'_L -Komponente ab und gibt diese Informationen I_i und X_i sowie die oben erwähnten Komponenten e_i und d_i und die öffentliche Information p in den Multikomponenten-Gruppenrechner ein, in welchem Z' durch sequentielles d_i -faches Rechnen von X_i und e_i -faches Rechnen von T_i für die Werte i von 1 bis L erhalten wird;
- 10 er gibt y_L und die öffentlichen Informationen p und g in den Gruppenrechner zur y_L -fachen Berechnung von g ein, um dadurch W zu erhalten; und
- er gibt Z' und W in den Komparator ein, um zu prüfen, ob $W = Z'$.
- Wenn die beiden übereinstimmen, erkennt der Verifizierer, daß das Dokument $\{m_1, \dots, m_L\}$ von den L autorisierten
- 15 Unterzeichnern ordnungsgemäß unterzeichnet wurde.
- Auch bei diesem typischen Schema sind jede Unterzeichnervorrichtung, eine Benutzervorrichtung und ein Aufzeichnungs- oder Speichermedium in ähnlicher Weise aufgebaut.
- Während bei dem Voranstehenden gilt $ID'_i = (ID'_{i-1}, ID_i)$, ist es auch möglich, die Einstellung so vorzusuchen, daß $ID'_i = (ID'_{i-1}, I_i)$. Dies erspart der Verifizierervorrichtung die Mühe der Suche nach der Identifikationsinformation ID_i für die
- 20 öffentliche Information I_i .

Ausführungsbeispiel 2

- Wenn bei dem Überlagerungssignaturschema und der Block-Signaturverifikation dafür, die oben unter Bezugnahme
- 25 auf die Fig. 2 bis 6 beschrieben wurden, das von dem ($i=1$)-ten Unterzeichner entsprechend der Unterzeichnervorrichtung 30_1 zu unterzeichnende Dokument m_1 auf in gesetzt wird und die Dokumente m_2, \dots, m_L in den Unterzeichnervorrichtungen 30_2 bis 30_L alle leer gemacht werden, werden die Unterzeichner 1 bis L das Dokument m auf Mehrfachsignaturbasis unterzeichnen. Ein Ausführungsbeispiel dieses Falles wird nachfolgend beschrieben. Dabei geht die Beschreibung von der Verwendung des Schnorr-Schemas aus.
- 30 Der Systemaufbau bei diesem Ausführungsbeispiel ist der gleiche wie in Fig. 1A, und die Zentrale 100 ist ebenfalls identisch mit der in Fig. 2 gezeigten aufgebaut. Außerdem führt die Zentrale 100 dieselbe Verarbeitung wie bei dem ersten Ausführungsbeispiel aus und erzeugt öffentliche Information $\{p, q, g\}$ mittels der anfänglichen Informationseinstellungsverarbeitung und liefert sie an die Unterzeichnervorrichtungen 30_1 bis 30_L sowie die Verifizierervorrichtung 800.
- Die Verarbeitung für den Unterzeichner i zum Einschreiben in das System ist ebenso die gleiche wie im Fall des ersten
- 35 Ausführungsbeispiels, weshalb auch die Unterzeichnervorrichtung 30_i gleich der in Fig. 3 gezeigten ist. Der Unterzeichner i erzeugt die öffentliche Information I_i mit dieser Verarbeitung und sendet sie sowie die Einweg-Funktionen f_i und h_i und die Identifikationsinformation ID_i über den sicheren Kommunikationskanal 400 an die Zentrale 100, damit diese Information dort als die öffentliche Information $\{ID_i, I_i, f_i, h_i\}$ registriert wird. Zugleich hält der Unterzeichner i s_i als geheime Information im Speicher 33.
- 40 Die anderen Unterzeichnervorrichtungen führen die gleiche Verarbeitung aus, wenn sie sich in das System einschreiben.
- Bei diesem Ausführungsbeispiel wird die unterzeichnete Nachricht des Dokuments m , die von dem Unterzeichner i ausgegeben wird, durch $\{ID'_i, X'_i, m, y_i\}$ repräsentiert. Der Unterzeichner ($i-1$) sendet die zu unterzeichnende Nachricht, und der Unterzeichner i erzeugt seine Signatur der Nachricht und fügt sie dieser bei und sendet die unterzeichnete Nachricht an den nächsten Unterzeichner ($i+1$). Wenn L Unterzeichner nacheinander die Nachricht unterzeichnen, wird i
- 45 schrittweise um eins von 1 auf L erhöht und die nachfolgende Prozedur wiederholt. Bei diesem Ausführungsbeispiel wird der Unterzeichner ($L+1$) als Verifizierer betrachtet. In diesem Fall gilt $ID'_0 =$ leere Menge, $X'_0 =$ leere Menge und $y_0 = 0$.

50 (2-A) Verarbeitung beim Unterzeichner i zur Signaturerzeugung

- Fig. 7 zeigt eine Interaktionsfolge einer Nachricht, und Fig. 8 zeigt den funktionalen Aufbau der Unterzeichnervorrichtung 30_i . Nach Empfang der Nachricht $\{ID'_{i-1}, X'_{i-1}, m, y_{i-1}\}$ von dem Unterzeichner ($i-1$) führt der Unterzeichner i die folgende Signaturerzeugungsverarbeitung aus. In dem Speicher 33 sind die öffentliche Information $\{p, q, g\}$, die von
- 55 der Zentrale 100 erhalten wird, die geheime Zufallszahl s_i und die Identifikationsinformation ID_i gespeichert.
- Schritt S1: Der Unterzeichner i erzeugt die Zufallszahl r_1 mit dem Zufallsgenerator 310 und gibt sie in den Modulo-Exponentiator 320 zusammen mit den öffentlichen Informationen p und g ein, so daß X_i unter Verwendung der Funktion Φ gemäß der nachstehenden Gleichung (31) berechnet wird:

$$60 \quad X_i = \Phi(r_i, g) = g^{r_i} \bmod p \quad (31)$$

Schritt S2: Der Unterzeichner i verwendet den f_i -Funktionsrechner 330 und den h_i -Funktionsrechner 340 zur Berechnung der beiden Informationen e_i bzw. d_i mittels der nachstehenden Gleichungen

$$65 \quad e_i = f_i(X'_i, m) \quad (32)$$

$$d_i = h_i(X'_i, m) \quad (33)$$

$$e_i = f_i(X'_i, m'_i)$$

$$d_i = h_i(X'_i, m'_i)$$

- 5 berechnet werden, um für jedes i ($1 \leq i \leq L$) die Komponenten e_i und d_i zu erhalten. Der Verifizierer leitet dann die entsprechende öffentliche Information I_i aus der ID_i -Komponente in der Information ID'_L und die Information X_i aus der X'_L -Komponente ab und gibt diese Informationen I_i und X_i sowie die oben erwähnten Komponenten e_i und d_i und die öffentliche Information p in den Multikomponenten-Gruppenrechner ein, in welchem Z' durch sequentielles d_i -faches Rechnen von X_i und e_i -faches Rechnen von T_i für die Werte i von 1 bis L erhalten wird;
- 10 er gibt y_L und die öffentlichen Informationen p und g in den Gruppenrechner zur y_L -fachen Berechnung von g ein, um dadurch W zu erhalten; und
- er gibt Z' und W in den Komparator ein, um zu prüfen, ob $W \equiv Z'$.

- Wenn die beiden übereinstimmen, erkennt der Verifizierer, daß das Dokument $\{m_1, \dots, m_L\}$ von den L autorisierten
- 15 Unterzeichnern ordnungsgemäß unterzeichnet wurde.

Auch bei diesem typischen Schema sind jede Unterzeichnervorrichtung, eine Benutzervorrichtung und ein Aufzeichnungs- oder Speichermedium in ähnlicher Weise aufgebaut.

- Während bei dem Voranstehenden gilt $ID'_i = (ID'_{i-1}, ID_i)$, ist es auch möglich, die Einstellung so vorzusehen, daß $ID'_i = (ID'_{i-1}, I_i)$. Dies erspart der Verifizierervorrichtung die Mühe der Suche nach der Identifikationsinformation ID_i für die
- 20 öffentliche Information I_i .

Ausführungsbeispiel 2

- Wenn bei dem Überlagerungssignaturschema und der Block-Signaturverifikation dafür, die oben unter Bezugnahme
- 25 auf die Fig. 2 bis 6 beschrieben wurden, das von dem $(i=1)$ -ten Unterzeichner entsprechend der Unterzeichnervorrichtung 30_1 zu unterzeichnende Dokument m_1 auf in gesetzt wird und die Dokumente m_2, \dots, m_L in den Unterzeichnervorrichtungen 30_2 bis 30_L alle leer gemacht werden, werden die Unterzeichner 1 bis L das Dokument m auf Mehrfachsignaturbasis unterzeichnen. Ein Ausführungsbeispiel dieses Falles wird nachfolgend beschrieben. Dabei geht die Beschreibung von der Verwendung des Schnorr-Schemas aus.

- 30 Der Systemaufbau bei diesem Ausführungsbeispiel ist der gleiche wie in Fig. 1A, und die Zentrale 100 ist ebenfalls identisch mit der in Fig. 2 gezeigten aufgebaut. Außerdem führt die Zentrale 100 dieselbe Verarbeitung wie bei dem ersten Ausführungsbeispiel aus und erzeugt öffentliche Information $\{p, q, g\}$ mittels der anfänglichen Informationseinstellungsverarbeitung und liefert sie an die Unterzeichnervorrichtungen 30_1 bis 30_L , sowie die Verifizierervorrichtung 800.

- Die Verarbeitung für den Unterzeichner i zum Einschreiben in das System ist ebenso die gleiche wie im Fall des ersten
- 35 Ausführungsbeispiels, weshalb auch die Unterzeichnervorrichtung 30_i gleich der in Fig. 3 gezeigten ist. Der Unterzeichner i erzeugt die öffentliche Information I_i mit dieser Verarbeitung und sendet sie sowie die Einweg-Funktionen f_i und h_i und die Identifikationsinformation ID_i über den sicheren Kommunikationskanal 400 an die Zentrale 100, damit diese Information dort als die öffentliche Information $\{ID_i, I_i, f_i, h_i\}$ registriert wird. Zugleich hält der Unterzeichner i s_i als geheime Information im Speicher 33.

- 40 Die anderen Unterzeichnervorrichtungen führen die gleiche Verarbeitung aus, wenn sie sich in das System einschreiben.

- Bei diesem Ausführungsbeispiel wird die unterzeichnete Nachricht des Dokuments m , die von dem Unterzeichner i ausgegeben wird, durch $\{ID'_i, X'_i, m, y_i\}$ repräsentiert. Der Unterzeichner $(i-1)$ sendet die zu unterzeichnende Nachricht, und der Unterzeichner i erzeugt seine Signatur der Nachricht und fügt sie dieser bei und sendet die unterzeichnete Nachricht an den nächsten Unterzeichner $(i+1)$. Wenn L Unterzeichner nacheinander die Nachricht unterzeichnen, wird i schrittweise um eins von 1 auf L erhöht und die nachfolgende Prozedur wiederholt. Bei diesem Ausführungsbeispiel wird der Unterzeichner $(L+1)$ als Verifizierer betrachtet. In diesem Fall gilt $ID'_0 =$ leere Menge, $X'_0 =$ leere Menge und $y_0 = 0$.

50 (2-A) Verarbeitung beim Unterzeichner i zur Signaturerzeugung

- Fig. 7 zeigt eine Interaktionsfolge einer Nachricht, und Fig. 8 zeigt den funktionalen Aufbau der Unterzeichnervorrichtung 30_i . Nach Empfang der Nachricht $\{ID'_{i-1}, X'_{i-1}, m, y_{i-1}\}$ von dem Unterzeichner $(i-1)$ führt der Unterzeichner i die folgende Signaturerzeugungsverarbeitung aus. In dem Speicher 33 sind die öffentliche Information $\{p, q, g\}$, die von
- 55 der Zentrale 100 erhalten wird, die geheime Zufallszahl s_i und die Identifikationsinformation ID_i gespeichert.

Schritt S1: Der Unterzeichner i erzeugt die Zufallszahl r_i mit dem Zufallsgenerator 310 und gibt sie in den Modulo-Exponentiator 320 zusammen mit den öffentlichen Informationen p und g ein, so daß X_i unter Verwendung der Funktion Φ gemäß der nachstehenden Gleichung (31) berechnet wird:

$$60 \quad X_i = \Phi(r_i, g) = g^{r_i} \bmod p \quad (31)$$

Schritt S2: Der Unterzeichner i verwendet den f_i -Funktionsrechner 330 und den h_i -Funktionsrechner 340 zur Berechnung der beiden Informationen e_i bzw. d_i mittels der nachstehenden Gleichungen

$$65 \quad e_i = f_i(X'_i, m) \quad (32)$$

$$d_i = h_i(X'_i, m) \quad (33)$$

wobei $X'_i = (X'_{i-1}, X_i)$.

Schritt S3: Der Unterzeichner i gibt diese Informationen e_i , d_i und r_i in den Modulo-Exponentiator **350** und dann in den Modulo-Addierer **360** zusammen mit der öffentlichen Information q und der geheimen Information s_i ein, um dadurch mit der Signaturfunktion Sg_i y_i wie folgt zu erzeugen:

$$y_i = Sg_i(e_i, d_i, s_i, r_i, y_{i-1}) = (y_{i-1} + d_i r_i + e_i s_i) \bmod q \quad (34).$$

Schritt S4: Der Unterzeichner i setzt $ID'_i = (ID'_{i-1}, ID_i)$ und sendet die Nachricht $\{ID'_i, X'_i, m, y_i\}$ an den nächsten Unterzeichner $(i+1)$.

(2-B) Verarbeitung beim Verifizierer zur Signaturverifikation

Fig. 9 zeigt den funktionalen Aufbau der Verifizierervorrichtung **800**. Wenn der Verifizierer die Nachricht $\{ID'_L, X'_L, m, y_L\}$ von dem Unterzeichner L empfängt, verifiziert er die Gültigkeit jeder Signatur mittels der nachstehend beschriebenen Verarbeitung.

Schritt S5: Der Verifizierer **800** bildet X'_i durch die ersten i Komponenten der Information X'_L und gibt sie und das Dokument m in den f_i -Funktionsrechner **810** und den h_i -Funktionsrechner **820** ein, worin die Komponenten e_i bzw. d_i mittels der nachstehenden Gleichungen berechnet werden:

$$e_i = f_i(X'_i, m) = f_i(X_1, \dots, X_i, m) \quad (35)$$

$$d_i = h_i(X'_i, m) = h_i(X_1, \dots, X_i, m) \quad (36)$$

Schritt S6: Der Verifizierer **800** leitet Information I_i von der ID'_i -Komponente in der Information ID'_L ab und extrahiert die X_i -Komponente in der Information X'_L und gibt diese Komponenten in den Multikomponenten-Modulo-Exponentiator **830** zusammen mit den Komponenten e_i und d_i und der öffentlichen Information p ein, so daß Z' mit der Verifikationsfunktion V gemäß nachstehender Gleichung berechnet wird:

$$Z' = V(X_i d_i, (I_i e_i) \mid i = 1, \dots, L) = X_1 d_1 I_1 e_1 \dots X_L d_L I_L e_L \bmod p \quad (37)$$

Schritt S7: Der Verifizierer **800** gibt die Information y_L zusammen mit der öffentlichen Information $\{p, g\}$, die im Speicher **88** gespeichert ist, in den Modulo-Exponentiator **840** ein, um W mit der Funktion Γ in folgender Weise zu berechnen:

$$W = \Gamma(y_L * g) = g^{y_L} \bmod p \quad (38)$$

Schritt S8: Der Verifizierer **800** gibt Z' und W in einen Komparator **850** ein, wo sie miteinander verglichen werden um zu sehen, ob $W = Z'$.

Wenn diese Bedingung erfüllt ist, wird davon ausgegangen, daß das Dokument in von den L autorisierten Unterzeichnern ordnungsgemäß unterzeichnet wurde.

Ein verbesserter Quadrier- und Multiplizier-Algorithmus zur Berechnung von $x^a y^b \bmod N$ in dem Multikomponenten-Modulo-Exponentiator kann der gleiche wie der zuvor in Verbindung mit dem ersten Ausführungsbeispiel beschriebene sein.

Unter Berücksichtigung des Weges der Erzeugung von y_L gilt:

$$g^{y_L} = g^{y_{L-1}} (g^{r_L})^{d_L} (g^{s_L})^{e_L} = g^{y_{L-1}} X_L d_L I_L e_L = \dots = X_1 d_1 I_1 e_1 \dots X_L d_L I_L e_L \bmod p \quad (39).$$

Wenn somit das Dokument in den oben genannten Test durch den Komparator **850** besteht, akzeptiert die der Verifizierer **800** das Dokument in als ordnungsgemäß von den L autorisierten Unterzeichnern unterzeichnet.

Es ist auch möglich, das System so auszugestalten, daß der Unterzeichner i vor seiner Signaturerzeugung prüft, ob die empfangene Nachricht $\{ID'_{i-1}, X'_{i-1}, m, y_{i-1}\}$ von den vorhergehenden Unterzeichnern 1 bis $(i-1)$ ordnungsgemäß unterzeichnet wurde, und, wenn dies der Fall ist, seine Signatur an der verifizierten Nachricht anbringt. In diesem Fall kann die Verifikation in gleicher Weise wie in den Schritten S10 bis S13 bei dem Verifizierer **800** ausgeführt werden. Dabei wird L in den Schritten S10 bis S13 durch $(i-1)$ ersetzt.

Die Unterzeichnervorrichtungen und die Verifizierervorrichtung führen die oben beschriebene Verarbeitung in der Regel mit Hilfe von Computern aus.

Wie zuvor erwähnt, ist dieses zweite Ausführungsbeispiel nicht nur auf die Schnorr-Schemata anwendbar, sondern auch auf die Fiat-Shamir-Schemata und digitale Signaturschemata, die die interaktiven Prüfungen einschließlich der Fiat-Shamir-Schemata verwenden. Demgemäß kann das Verfahren gemäß der vorliegenden Erfindung allgemein wie folgt zusammengefaßt werden:

Die Systemparameter, die veröffentlicht werden, sind p zur Angabe der Anzahl von Elementen der Gruppe, ein Element g der Gruppe, mit dem eine Gruppenberechnung beginnt, und eine positive ganze Zahl q derart, daß, wenn das Element g q -mal berechnet wird, die Rechnung zu dem Element g zurückkehrt.

Der Unterzeichner i erzeugt die Zufallszahl s_i mittels des Zufallsgenerators bei seiner Einschreibung in das System und gibt die Zufallszahl s_i und die öffentlichen Informationen g und p in einen Gruppenrechner ein, in welchem das Element g s_i -mal gerechnet wird, um die öffentliche Information I_i zu berechnen; und veröffentlicht die öffentliche Information I_i und die Einweg-Funktionen f_i und h_i zusammen mit der Identifikationsinformation ID_i , behält jedoch die Zufallszahl s_i als geheime Information.

Bei der Signaturerzeugungsverarbeitung nach Empfang der unterzeichneten Nachricht $\{ID'_{i-1}, X'_{i-1}, m, y_{i-1}\}$ von dem Unterzeichner $(i-1)$ auf der Basis der Nachricht m , macht der Unterzeichner i folgendes:

- er erzeugt die Zufallszahl r unter Verwendung des Zufallsgenerators, gibt sie dann zusammen mit den öffentlichen Informationen p und g in den Gruppenrechner ein, um das Element $g \cdot r_i$ -mal zum Erhalt der Information X_i zu rechnen, und
 5 setzt $X'_i = (X'_{i-1}, X_i)$;
 er berechnet die Komponenten e_i und d_i durch

$$e_i = f_i(X'_i, m)$$

$$10 \quad d_i = h_i(X'_i, m)$$

unter Verwendung des f_i -Funktionsrechners und des h_i -Funktionsrechners; und

- er gibt die Informationen e_i , d_i und r_i zusammen mit der öffentlichen Information q und der geheimen Information s_i in einen Exponentialkomponenten-Multiplizierer und Exponentialkomponenten-Addierer ein, worin y_i mit der Signaturfunktion Sg_i errechnet wird, und zwar durch

$$15 \quad y_i = Sg_i(e_i, d_i, s_i, r_i, y_{i-1}) = (y_{i-1} + d_i r_i + e_i s_i) \bmod q,$$

und er setzt dann $ID'_i = (ID'_{i-1}, ID_i)$ und sendet die Nachricht $\{ID'_i, X'_i, m, y_i\}$ an den nächsten Unterzeichner $(i+1)$.

- 20 Andererseits bildet der Verifizierer 800, wenn er die Nachricht $\{ID'_L, X'_L, m, y_L\}$ von dem Unterzeichner L empfängt, X'_i aus den ersten i Komponenten der Information X'_L und gibt die Informationen X'_i und in dann in den f_i -Funktionsrechner und den h_i -Funktionsrechner ein, worin

$$e_i = f_i(X'_i, m)$$

$$25 \quad d_i = h_i(X'_i, m)$$

berechnet werden, um für jedes i die Komponenten e_i und d_i zu erhalten. Der Verifizierer leitet dann die entsprechende öffentliche Information I_i aus der ID_i -Komponente in der Information ID'_L und die Information X_i aus der X'_L -Komponente ab und gibt diese Informationen I_i und X_i sowie die oben erwähnten Komponenten e_i und d_i und die öffentliche Information p in den Multikomponenten-Gruppenrechner ein, in welchem Z' durch sequentielles d_i -faches Rechnen von X_i und e_i -faches Rechnen von T_i für die Werte i von 1 bis L erhalten wird;

- er gibt y_L und die öffentlichen Informationen p und g in den Gruppenrechner zur y_L -fachen Berechnung von g ein, um dadurch W zu erhalten; und
 35 er gibt Z' und W in den Komparator ein, um zu prüfen, ob $W \equiv Z'$.

Wenn die beiden übereinstimmen, erkennt der Verifizierer, daß das Dokument m von den L autorisierten Unterzeichnern ordnungsgemäß unterzeichnet wurde.

Auch bei diesem typischen Schema sind die einzelnen Unterzeichnervorrichtungen, eine Benutzervorrichtung und ein Aufzeichnungs- oder Speichermedium in ähnlicher Weise aufgebaut.

- 40 Während bei dem Voranstehenden gilt $ID'_i = (ID'_{i-1}, ID_i)$, ist es auch möglich, die Einstellung so vorzusehen, daß $ID'_i = (ID'_{i-1}, I_i)$. Dies erspart dem Verifizierer die Mühe der Suche nach der Identifikationsinformation ID_i für die öffentliche Information I_i .

Ausführungsbeispiel 3

- 45 Als nächstes wird in Verbindung mit dem Fall des Einsatzes des Schnorr-Schemas ein Ausführungsbeispiel beschrieben, bei dem in dem System von Fig. 1B die Unterzeichner entsprechend den Unterzeichnervorrichtungen 30₁ bis 30_L einzeln ihre Signatur an jeweiligen Dokumenten m_1 bis m_L anbringen und sie an die Verifizierervorrichtung 800 liefern, damit die unterzeichneten Dokumente dort en-bloc verifiziert werden. Die Idee der Verwendung der zweiten Exponentialkomponente, die unten beschrieben wird, ist ebenso in weitem Umfang auf die Fiat-Shamir-Schemata und digitale Signaturschemata anwendbar, die diese enthaltende interaktive Prüfungen verwenden.

(3-A) Anfängliche Informationseinstellungsverarbeitung

- 55 Die Zentrale 100 stimmt in ihrer Ausgestaltung mit derjenigen von Fig. 2 hinsichtlich der anfänglichen Informationseinstellungsverarbeitung überein, bei der die Werte $\{p, q, g\}$, die für das System einzigartig sind, veröffentlicht werden, weshalb die folgende Verarbeitung auch die gleiche wie die im Fall von Fig. 2 ist:

Schritt S1: Die Zentrale 100 erzeugt die Primzahl p mittels des Primzahlgenerators 110 und die Primzahl q , die ein Maß für $p-1$ ist, mittels des Dividierers 120.

- 60 Schritt S2: Die Zentrale 100 erzeugt das Grundelement α von $(Z/pZ)^*$ mittels des Grundelementgenerators 130 und die ganze Zahl g der Ordnung bzw. des Grads q als den vorher erwähnten Parameter β durch die nachfolgende Berechnung unter Verwendung des Modulo-Exponentiators 140, der die Funktion G_1 berechnet, die zuvor in Verbindung mit den Prinzipien der vorliegenden Erfindung beschrieben wurde:

$$65 \quad \beta = g = G_1(q) = \alpha^{(p-1)/q} \bmod p \quad (40)$$

Schritt S3: Die öffentliche Information $\{p, q, g\}$ wird an die Unterzeichnervorrichtungen 30₁, ..., 30_L und die Verifizierervorrichtung 800 über die sicheren Kanäle 400 geschickt.

(3-B) Verarbeitung durch den Unterzeichner i im Moment des Anschließens an das System

Als nächstes erfolgt unter Bezugnahme auf Fig. 10 eine Beschreibung der Verarbeitung, die beim Unterzeichner i für sein Einschreiben bei dem System ausgeführt wird. Es wird angemerkt, daß in dem Speicher 33 die öffentliche Information {p, q, g} gespeichert ist, die von der Zentrale 100 empfangen wird.

Schritt S4: Der Unterzeichner i erzeugt eine Zufallszahl s_i mittels des Zufallsgenerators 31 und gibt sie sowie die öffentlichen Informationen $g (= \beta)$ und p in den Modulo-Exponentiator 32 ein, der die Funktion $G_2(s_i, \beta)$ berechnet, auf die zuvor in Verbindung mit den Prinzipien der Erfindung verwiesen wurde. Auf diese Weise wird zum Erhalt der öffentlichen Information I_i die folgende Berechnung ausgeführt:

$$I_i = G_2(s_i, g) = g^{s_i} \bmod p \quad (41)$$

Schritt S5: Der Unterzeichner i sendet die Identifikationsinformation ID_i , die öffentliche Information I_i und die Einwegfunktionen f_i und h_i über den sicheren Kanal 400 an die Zentrale 100, wo sie als öffentliche Information registriert werden. Zugleich behält der Unterzeichner i die Zufallszahl s_i als geheime Information im Speicher 33.

Bei den anderen Unterzeichnern wird dieselbe Verarbeitung ausgeführt, wenn sie sich bei dem System einschreiben. In der folgenden Beschreibung wird das unterzeichnete Dokument mit $\{ID_i, X_i, m_i, y_i\}$ identifiziert, und zwar in der Annahme, daß der Unterzeichner i das Dokument m_i unterzeichnet.

(3-C) Verarbeitung beim Unterzeichner i zur Signaturerzeugung

Fig. 11 zeigt Interaktionsfolgen von Nachrichten, und Fig. 12 zeigt den funktionalen Aufbau der Unterzeichnervorrichtung 30i.

Schritt S6: Der Unterzeichner i (Unterzeichnervorrichtung 30i) erzeugt die Zufallszahl r_i mittels des Zufallsgenerators 310 und gibt sie in den Modulo-Exponentiator 320 ein, der die Funktion Φ berechnet, und zwar zusammen mit den öffentlichen Informationen p und g . X_i wird auf folgende Weise errechnet:

$$X_i = \Phi(r_i, g) = g^{r_i} \bmod p \quad (42)$$

Schritt S7: Der Unterzeichner i verwendet den f_i -Funktionsrechner 330 und den h_i -Funktionsrechner 340 zur Berechnung der beiden Informationen e_i bzw. d_i aufgrund der nachstehenden Gleichungen

$$e_i = f_i(X_i, m_i) \quad (43)$$

$$d_i = h_i(X_i, m_i) \quad (44).$$

Schritt S8: Der Unterzeichner i gibt diese Informationen e_i , d_i und r_i in den Modulo-Exponentiator 350 und dann in den Modulo-Addierer 360 zusammen mit der öffentlichen Information q und der geheimen Information s_i ein, um mit der Signaturfunktion Sg_i in nachstehender Weise y_i zu berechnen:

$$y_i = Sg_i(e_i, d_i, s_i, r_i, q) = (d_i r_i + e_i s_i) \bmod q \quad (45).$$

Schritt S9: Der Unterzeichner i sendet die Nachricht $\{ID_i, X_i, m_i, y_i\}$ an den Verifizierer 800.

(3-D) Prozeß beim Verifizierer zur Signaturverifikation

Fig. 13 zeigt den funktionalen Aufbau der Verifizierervorrichtung 800. In dem Speicher 88 ist die öffentliche Information {p, q, g} gespeichert, die von der Zentrale erhalten wird. Wenn er die L Nachrichten $\{ID_i, X_i, m_i, y_i\}$ von den L Unterzeichnern empfängt, führt der Verifizierer 800 die nachfolgende Verarbeitung aus, um die jeweiligen Signaturen en bloc zu verifizieren.

Schritt S10: Der Verifizierer 800 gibt die Information X_i und das Dokument m_i in den f_i -Funktionsrechner 810 und den h_i -Funktionsrechner 820 ein, in denen die Komponenten e_i bzw. d_i ($1 \leq i \leq L$) jeweils berechnet werden durch

$$e_i = f_i(X_i, m_i)$$

$$d_i = h_i(X_i, m_i).$$

Schritt S11: Der Verifizierer 800 empfängt von der Zentrale 100 die öffentliche Information I_i entsprechend der Identifikationsinformation ID_i und gibt die öffentliche Information I_i zusammen mit den Komponenten e_i und d_i , die in oben beschriebener Weise erzeugt wurden, und der öffentlichen Information p in den Multikomponenten-Modulo-Exponentiator 830 ein, wo Z' gemäß nachfolgender Gleichung (46) berechnet wird

$$Z' = V(X_i * d_i, (I_i * e_i) \mid i = 1, \dots, L) = (X_1^d_1 I_1^{e_1} \dots X_L^d_L I_L^{e_L}) \bmod p \quad (46).$$

Schritt S12: Der Verifizierer 800 gibt L Informationen y_i (y_1 bis y_L) und die öffentliche Information q in den Modulo-Addierer 840 ein, um auf folgende Weise einen akkumulierten Wert Y zu berechnen

$$Y = \sum_{i=1}^L y_i \bmod q. \quad (47)$$

5 Dann gibt der Verifizierer **800** Y und die öffentliche Information $\{p, q\}$ in den Modulo-Exponentiator **845** ein, der die Funktion $\Gamma(Y * g)$ berechnet. W wird auf folgende Weise berechnet

$$W = \Gamma(Y * g) = g^Y \bmod p \quad (48).$$

10 Schritt S13: Z' und W werden einem Komparator **850** eingegeben, wo sie verglichen werden um festzustellen, ob $W = Z'$.

Wenn dies der Fall ist, wird davon ausgegangen, daß das Dokument in von dem jeweiligen der L autorisierten Unterzeichner ordnungsgemäß unterzeichnet wurde.

15 Eines der Verfahren der Quadrier- und Multiplizier-Algorithmen zur Berechnung von $x^a y^b \bmod N$ in dem Multikomponenten-Modulo-Exponentiator **830** ist das gleiche wie das voranstehend beschriebene.

Unter Berücksichtigung der Art der Erzeugung des akkumulierten Werts Y gilt

$$g^Y = g^{y_{L-1}} (g^{r_{L-1}})^{d_{L-1}} (g^{s_{L-1}})^{e_{L-1}} = g^{y_{L-1}} X_{L-1}^{d_{L-1}} I_{L-1}^{e_{L-1}} = \dots = X_1^{d_1} \dots X_L^{d_L} I_1^{e_1} \dots I_L^{e_L} \bmod p \quad (49).$$

20 Wenn somit die Dokumente m_i den obigen Test mittels des Komparators **860** passieren, akzeptiert der Verifizierer **800** die Dokumente als von den L autorisierten oder gültigen Unterzeichnern ordnungsgemäß unterzeichnet.

Es erfolgt nun eine Beschreibung der Verarbeitung in dem Fall, wo $W = Z'$ in Schritt S13 nicht erfüllt ist. Beispielsweise werden, wenn $L = 100$, die Nachrichten $\{ID_i, X_i, m_i, y_i\}$ in zwei Gruppen ($L/2 = 50$) unterteilt, und die Verarbeitung der Schritte S9 bis S13 wird für die Nachrichten einer der beiden Gruppen ausgeführt um festzustellen, ob eine Fehlüber einstimmung zwischen ihnen gefunden wird. Wenn eine Fehlüber einstimmung gefunden wird, werden die Nachrichten dieser Gruppe weiter in zwei Untergruppen unterteilt. Wird keine Fehlüber einstimmung gefunden, werden die Nachrichten der anderen Gruppe in zwei Untergruppen unterteilt. Dann werden die Nachrichten einer der beiden Untergruppen der Verarbeitung der Schritte S9 bis S13 unterzogen. Durch Wiederholen dieser Verarbeitung ist der Verifizierer in der Lage, den Unterzeichner zu identifizieren, der das betroffene Dokument nicht ordnungsgemäß unterzeichnet hat.

30 Wie zuvor erwähnt, ist die vorliegende Erfindung nicht nur auf die Schnorr-Schemata, sondern auch auf die Fiat-Shamir-Schemata und digitale Signaturschemata anwendbar, die die interaktiven Prüfungen unter Einschluß der Fiat-Shamir-Schemata verwenden. Demgemäß kann das Verfahren der vorliegenden Erfindung allgemein wie folgt zusammengefaßt werden:

35 Die Systemparameter, die veröffentlicht werden, sind p zur Spezifizierung der Anzahl von Elementen in der Gruppe, ein Element g der Gruppe, mit der die Gruppenberechnung beginnt, und eine positive ganze Zahl q derart, daß, wenn das Element g q -mal berechnet wird, die Rechnung zum Element g zurückkehrt.

Der Unterzeichner i erzeugt die Zufallszahl s_i mittels des Zufallsgenerators zum Zeitpunkt des Einschreibens in das System und gibt die Zufallszahl s_i und die öffentlichen Informationen g und p in einen Gruppenrechner ein, wo das Element g s_i -mal berechnet wird, um die öffentliche Information I_i zu berechnen, und

40 veröffentlicht die öffentliche Information I_i ; die Funktionen f_i und h_i zusammen mit der Identifikationsinformation ID_i , behält jedoch die Zufallszahl s_i als geheime Information.

Bei der Signaturerzeugungsverarbeitung führt der Unterzeichner i folgendes aus:
er erzeugt die Zufallszahl r_i unter Verwendung des Zufallsgenerators und gibt sie dann zusammen mit den öffentlichen

45 Informationen p und g in den Gruppenrechner ein, um das Element g r_i -mal zu berechnen und die Information X_i zu erhalten;

er berechnet die Komponenten e_i und d_i gemäß

$$e_i = f_i(X_i, m_i)$$

$$50 \quad d_i = h_i(X_i, m_i)$$

unter Verwendung des f_i -Funktionsrechners und des h_i -Funktionsrechners; und

er gibt die Informationen e_i , d_i und r_i zusammen mit der öffentlichen Information q und der geheimen Information s_i in einen Exponentialkomponenten-Multiplizierer und einen Exponentialkomponenten-Addierer ein, um y_i in folgender

$$y_i = (d_i r_i + e_i s_i) \bmod q$$

60 und erhält auf diese Weise die Nachrichten $\{ID_i, X_i, m_i, y_i\}$ und sendet sie an den Verifizierer.

Wenn er die Nachrichten $\{ID_i, X_i, m_i, y_i\}$ ($1 \leq i \leq L$) von den L Unterzeichnern empfängt, gibt der Verifizierer die Information X_i und die Nachricht m_i in den f_i -Funktionsrechner und den h_i -Funktionsrechner ein, in denen

$$e_i = f_i(X_i, m_i)$$

$$65 \quad d_i = h_i(X_i, m_i)$$

berechnet werden, um die Komponenten e_i und d_i für alle Werte von i zu erhalten. Der Verifizierer leitet dann die öffent-

liche Information I_i aus der Identifikationsinformation ID_i ab und gibt diese Informationen I_i und X_i sowie die oben erwähnten Komponenten e_i und d_i und die öffentliche Information p in den Multikomponenten-Gruppenrechner ein, worin Z' durch d_i -maliges sequentielles Berechnen von X_i und e_i -maliges Berechnen von T_i für die Werte i von 1 bis L erhalten wird.

Der Verifizierer gibt L Informationen y und die öffentliche Information p in den Exponentialkomponenten-Addierer ein, um Y auf folgende Weise zu berechnen

$$Y = \sum_{i=1}^L y_i \text{ mod } q \quad 10$$

und gibt dann Y und die öffentlichen Informationen p und g in einen Gruppenrechner ein, wo g zum Erhalt von W Y -mal berechnet wird.

Z' und W werden in den Komparator eingegeben um zu prüfen, ob $W \equiv Z'$ und, wenn beide übereinstimmen erkennt der Verifizierer, daß die L Dokumente m_i von den L autorisierten Unterzeichnern ordnungsgemäß unterzeichnet wurden. 15

Die Unterzeichnervorrichtungen und die Verifizierervorrichtung führen die Verarbeitung gewöhnlich mittels eines Computers aus.

Während bei den Ausführungsbeispielen 1 bis 3 die Verwendung von Z_q als kommutative Gruppe des endlichen Feldes beschrieben wurde, das durch den Parameter q definiert ist, kann eine elliptische Kurve als die kommutative Gruppe verwendet werden. Dies löst das Problem der Zunahme der von dem Unterzeichner zur Signaturzeugung zu verarbeitenden Rechenmenge. Das RSA-Verschlüsselungssystem gründet seine Sicherheit auf der Schwierigkeit des (eingangs erwähnten) Faktorzerlegungsproblems, während die Sicherheit des sogenannten Ellipsen-Verschlüsselungssystem auf einem diskreten Logarithmenproblem an einer elliptischen Kurve beruht, dessen Lösung für schwieriger gehalten wird als die des Faktorzerlegungsproblems. 20

Das Folgende ist eine Definition einer elliptischen Kurve auf dem endlichen Feld $GF(q)$, das durch die Verwendung der Parameter $a, b \in GF(q)(4a^3 + 27b^2 \neq 0)$ gegeben ist: 25

$$E_{a,b}(GF(q)) = \{(x, y) \in GF(q)^2 | y^2 = x^3 + ax + b\} \cup \{0\} \quad (50)$$

worin $GF(q)$ ein Definitionsfeld der elliptischen Kurve $E_{a,b}(GF(q))$ genannt wird und 0 einen unendlichen Punkt angibt. 30

Die Addition auf der elliptischen Kurve ist beispielsweise wie folgt:
Wenn man setzt

$$P_i = (x_i, y_i) \in E_{a,b}(GF(q)), \quad 35$$

(wobei $i = 1, 2$), kann $(x_3, y_3) = P_1 + P_2$ wie folgt geschrieben werden:

(a) wenn $P_1 \neq P_2$, und $\lambda = (y_2 - y_1)/(x_2 - x_1)$ gesetzt wird,

$$x_3 = \lambda^2 - (x_1 + x_2), y_3 = -y_1 + \lambda(x_1 - x_3) \quad (51) \quad 40$$

(b) wenn $P_1 = P_2$, wenn also $(x_3, y_3) = 2P_1$, und $\lambda = (3x_1^2 + a)/(2y_1)$ gesetzt wird,

$$x_3 = \lambda^2 - 2x_1, y_3 = -y_1 + \lambda(x_1 - x_3) \quad (52). \quad 45$$

Die Gruppenberechnung auf der elliptischen Kurve ist beispielsweise beschrieben in D. R. Stinson, "CRYPTOGRAPHY Theory and Practice", CRC Press, Seiten 187–190, 1995, In der folgenden Beschreibung wird die Addition von P_1 und P_2 auf der elliptischen Kurve $E_{a,b}(GF(q))$ durch

$$(P_1 + P_2) \text{ über } E_{a,b}(GF(q)) \quad 50$$

dargestellt.

Da die gegenwärtig bekannte Lösung des diskreten Logarithmenproblems auf der elliptischen Kurve weniger effizient und schwieriger als die Lösung des Faktorzerlegungsproblems ist, ist es möglich, den Parameter g des Definitionsfeldes der elliptischen Kurve klein zu machen und dementsprechend die beinhaltete Rechenkomplexität zu verringern. Konkret sagt man, daß dieselbe Sicherheit wie im Fall von $|N| = 1024$ durch $|q| = 160$ garantiert werden kann, wobei $|q|$, wie schon oben definiert, die Anzahl Bits der Primzahl p repräsentiert (siehe beispielsweise Bruce Schneier, "APPLIED CRYPTOGRAPHY (Second Edition)", John Wiley & Sons, Inc., Seiten 480–481, 1996). 55

Das gewöhnliche diskrete Logarithmenproblem besteht darin, daß, wenn eine ganze Zahl $g \in (Z/pZ)^* = \{1, 2, \dots, p-1\}$, wobei p eine große Primzahl ist und g die Ordnung q hat, als öffentliche Information geliefert wird, $y \in Z/qZ$ zu berechnen ist, für das $g^y \equiv x \pmod{p}$ bezüglich einer ganzen Zahl $x \in (Z/pZ)^*$ erfüllt ist. 60

Andererseits besteht das diskrete Logarithmenproblem auf der elliptischen Kurve darin, daß, wenn das Definitionsfeld $GF(q)$, die Parameter a und b der elliptischen Kurve und ein Punkt $P \in E_{a,b}(GF(q))$ einer Ordnung k auf der elliptischen Kurve als öffentliche Information geliefert werden, $y \in Z/kZ$ zu berechnen ist, das $yP \equiv X$ über $E_{a,b}(GF(q))$ bezüglich eines Punkts X auf der elliptischen Kurve erfüllt. 65

Der Punkt P wird Basispunkt genannt. $yP \equiv X$ gibt an, daß der Basispunkt P , wenn er auf der elliptischen Kurve y -mal addiert wird, mit dem Punkt $X \in E_{a,b}(GF(q))$ zusammenfällt. Die y -fache Addition des Basispunkts P auf der elliptischen Kurve $E_{a,b}(GF(q))$ ist speziell durch yP über $E_{a,b}(GF(q))$ repräsentiert, das zur Definition einer Gruppenrechnung auf der

elliptischen Kurve verwendet wird.

Bei Verwendung der oben definierten Gruppenrechnung auf der elliptischen Kurve, könnten ein sogenanntes Diffie-Hellman-Key-Sharing-Schema, ein ElGamal-Verschlüsselungssystem und ein ElGamal-Signaturschema, die die Schwierigkeit des gewöhnlichen diskreten Logarithmenproblems nutzen, alle zu Schemata modifiziert werden, die die Schwierigkeit des diskreten Logarithmus auf der elliptischen Kurve nutzen.

Die Schnorr- und die Fiat-Shamir-Schemata, die interaktive Prüfungen verwenden, könnten ebenfalls zu Schemata modifiziert werden, die die Schwierigkeit des diskreten Logarithmus auf der elliptischen Kurve nutzen. Eine Beschreibung erfolgt beispielsweise für eine digitale Signatur mittels des Schnorr-Schemas unter Einsatz der elliptischen Kurve.

Eine vertrauenswürdige Institution (vertrauenswürdige Zentrale) veröffentlicht den Parameter q des Definitionsfeldes $\text{GF}(q)$, den Parameter $a, b, \text{GF}(q)$ der elliptischen Kurve und den Basispunkt $P \in E_{a,b}(\text{GF}(q))$ einer Ordnung bzw. eines Grads k auf der elliptischen Kurve.

Schritt 1: Ein Unterzeichner A erzeugt eine Zufallszahl $s \in (\mathbb{Z}/k\mathbb{Z})$ und berechnet öffentliche Information I durch

$$I = sP \text{ über } E_{a,b}(\text{GF}(q)) \quad (53)$$

und veröffentlicht ein Paar aus Identifikationsinformationen (ID) und Information I.

Der Unterzeichner A durchläuft die folgende Prozedur, um einem Verifizierer B zu beweisen, daß ein Dokument m echt ist.

Schritt 2: Der Unterzeichner A erzeugt eine Zufallszahl $r \in (\mathbb{Z}/k\mathbb{Z})$ und berechnet

$$X = rP \text{ über } E_{a,b}(\text{GF}(q)) \quad (54).$$

Schritt 3: Der Unterzeichner A berechnet eine ganze Zahl $e \in (\mathbb{Z}/k\mathbb{Z})$ unter Verwendung einer Einweg-Funktion f durch

$$e = f(X, m) \quad (55).$$

Schritt 4: Der Unterzeichner A erzeugt eine Signatur y durch

$$y = (r + es) \bmod k \quad (56)$$

und sendet $\{ID, m, X, y\}$ als eine unterzeichnete Nachricht an den Verifizierer B.

Schritt 5: Der Verifizierer B berechnet die ganze Zahl $e \in (\mathbb{Z}/k\mathbb{Z})$ unter Verwendung der Einweg-Funktion f durch

$$e = f(X, m).$$

Schritt 6: Der Verifizierer B prüft, ob

$$yP = (X + eI) \text{ über } E_{a,b}(\text{GF}(q)) \quad (57)$$

erfüllt ist, wobei I öffentliche Information entsprechend der Identifikationsinformation ID ist.

Unter Berücksichtigung des Wegs zur Erzeugung von y gilt

$$yP = (r + es)P = rP + e(sP) = (X + eI) \text{ über } E_{a,b}(\text{GF}(q)) \quad (58).$$

Wenn somit Gleichung (57) erfüllt ist, erkennt der Verifizierer B, daß das Dokument m von dem Unterzeichner A ordnungsgemäß unterzeichnet ist.

Bei dem Voranstehenden könnte die Signatur des Unterzeichners A gefälscht werden, falls $\{ID, X, m, y\}$ als unterzeichnete Nachricht gesendet würde, wenn die ganze Zahl $e \in (\mathbb{Z}/k\mathbb{Z})$, für die $e = f(X, m)$ gilt, durch Berechnung von $X \in E_{a,b}(\text{GF}(q))$ ermittelt werden könnte, welches die Verifikationsgleichung erfüllt, nachdem die ganzen Zahlen $e \in (\mathbb{Z}/k\mathbb{Z})$ und $y \in (\mathbb{Z}/k\mathbb{Z})$ geeignet gewählt wurden. Da die Wahrscheinlichkeit, daß die Verifikationsgleichung $e = f(X, m)$ erfüllt ist, $1/k$ ist, hängt die Rechenkomplexität, die mit der Fälschung der Signatur verbunden ist, von dem Wert k ab.

Das elliptische Schnorr-Schema beinhaltet die Berechnung der Gleichungen (51) und (52) für eine n -fach-Punktbe-rechnung (einschließlich Modulo- q -Rechnungen) auf der elliptischen Kurve in einer mittleren Häufigkeit von $3|q|/2$, eine einzelne Multiplikation (einschließlich Modulo- k -Rechnungen) von $|k|$ Bit Ganzzahlen und eine einzelne Addition (einschließlich Modulo- k -Rechnungen) der $|k|$ Bit Ganzzahlen.

Das obige Verfahren mit der elliptischen Kurve wird nachfolgend in Anwendung auf die zuvor beschriebenen Ausführungsbeispiele 1 bis 3 beschrieben.

Das Ergebnis der Addition $P_3 (P_1 + P_2)$ auf der elliptischen Kurve wird durch die Gleichungen (51) und (52) unter Verwendung von x - und y -Koordinaten berechnet. Wie aus Gleichung (59) hervorgeht, die die elliptische Kurve definiert, ist, wenn die x -Koordinate einmal bestimmt ist, der Punkt auf der elliptischen Kurve eindeutig abhängig davon definiert, ob die y -Koordinate plus oder minus ist. Da die x -Koordinate der Wert des Definitionsfeldes $\text{GF}(q)$ ist, muß hier darauf hingewiesen werden, daß der Punkt auf der elliptischen Kurve durch $(|q| + 1)$ Bits repräsentiert werden kann.

Ausführungsbeispiel 4

Dieses Ausführungsbeispiel entspricht dem ersten Ausführungsbeispiel, das die Überlagerungssignatur und deren Block-Verifikation durchführt. Nachstehend erfolgt eine Beschreibung eines Ausführungsbeispiels, bei dem das Schnorr-

Schema auf die Überlagerungssignatur und deren Block-Verifikation angewendet wird und welches das Verfahren der elliptischen Kurve einsetzt.

Die Idee der Verwendung einer zweiten Mehrfachkomponente, die unten beschrieben wird, kann in weitem Umfang auf die ElGamal-Signatur-Schemata und die digitalen Signatur-Schemata, die unter deren Einschluß die interaktiven Prüfungen verwenden, angewendet werden.

Die Systemkonfiguration, auf die dieses Ausführungsbeispiel angewendet wird, ist die gleiche wie die in Fig. 1A gezeigte. Deren Beschreibung soll hier daher nicht wiederholt werden.

(4-A) Anfängliche Informationseinstellverarbeitung

Nachfolgend wird unter Bezugnahme auf Fig. 14 die anfängliche Informationseinstellverarbeitung beschrieben, die ausgeführt wird, wenn die Zentrale 100 das System startet. Diese Verarbeitung ist dazu gedacht, einen Wert $\{q, a, b, P, k\}$, der für das System einzigartig ist, zu veröffentlichen.

Schritt S1: Die Zentrale 100 erzeugt die Primzahl q mittels des Primzahlgenerators 110 und $a, b \in GF(q)$ mittels eines Parametergenerators 120.

Schritt S2: Die Zentrale 100 erzeugt einen Punkt $P \in E_{a,b}(GF(q))$ auf der elliptischen Kurve mittels eines Basispunktgenerators 130, und den Grad oder die Ordnung des Basispunkts mittels eines Ordnungsrechners 140. Die elliptische Kurve $E_{a,b}(GF(q))$ entspricht einem Wert der Funktion $G_1(q)$, auf die früher in Verbindung mit den Prinzipien der vorliegenden Erfindung Bezug genommen wurde, und der Punkt P entspricht dem dabei erwähnten β .

Schritt S3: Die öffentliche Information $\{g, a, b, P, k\}$ wird an die Unterzeichner (Unterzeichnervorrichtungen 30₁, ..., 30_L) und an den Verifizierer 800 über die sicheren Kommunikationskanäle 400 gesandt und in den Speichern 33 bzw. 88 gespeichert.

Der Ordnungs- oder Gradrechner 140 kann leicht beispielsweise unter Verwendung des Schoof-Algorithmus zur Berechnung der Ordnung oder des Grades der elliptischen Kurve $E_{a,b}(GF(q))$ (der Anzahl von Punkten auf der Kurve) implementiert werden, (siehe beispielsweise R. Schoof, "Elliptic Curves Over Finite Fields and the Computation of Square Roots Mod p ", Math. Com., 44, Seiten 483-494, 1985).

(4-B) Verarbeitung beim Unterzeichner i für dessen Einschreiben in dem System

Unter Bezugnahme auf Fig. 15 wird nun die Verarbeitung beschrieben, die bei einem Unterzeichner i durchgeführt wird, wenn er sich in dem System einschreibt.

Schritt S4: Der Unterzeichner i erzeugt die Zufallszahl s_i mittels des Zufallsgenerators 31 und gibt sie sowie die öffentliche Information $\{q, a, b, P\}$ in einen n -fach-Punktrechner 32 ein, in welchem die öffentliche Information I_i mit der zuvor erwähnten Funktion G_2 gemäß nachstehender Gleichung (59) berechnet wird

$$I_i = G_2(s_i, P) = s_i P \text{ über } E_{a,b}(GF(q)) \quad (59).$$

Schritt S5: Der Unterzeichner i sendet die Identifikationsinformation ID_i , die öffentliche Information I_i und die Einweg-Funktionen f_i und h_i über den sicheren Kommunikationskanal 400 an die Zentrale 100, damit diese als öffentliche Information $\{ID_i, I_i, f_i, h_i\}$ registriert werden. Der Unterzeichner i behält die Zufallszahl s_i als geheime Information in dem Speicher 33.

In der folgenden Beschreibung wird die unterzeichnete Version des Dokuments m'_i , die von dem Unterzeichner i geliefert wird, durch $\{ID'_i, X'_i, y_i\}$ identifiziert. Die Interaktionsfolge der Nachricht ist die gleiche wie die im Fall von Fig. 4. Bei Empfang einer Nachricht $\{ID'_{i-1}, X'_{i-1}, m'_{i-1}, y_{i-1}\}$ von dem Unterzeichner $(i-1)$, führt der Unterzeichner i die Signaturerzeugungsverarbeitung durch, die nachstehend beschrieben wird. Die Konfiguration der Unterzeichnervorrichtung 30_i des Unterzeichners i ist in Fig. 16 dargestellt. Es erfolgt nun eine Beschreibung des Falles, wo der Unterzeichner $(i-1)$ die zu unterzeichnende Nachricht sendet und der Unterzeichner i seine Signatur an der Nachricht anbringt und die unterzeichnete Nachricht an den nächsten Unterzeichner $(i+1)$ sendet. Im Fall der überlagerten Signatur durch L Unterzeichner braucht lediglich i schrittweise um 1 von 1 bis L erhöht zu werden und die folgende Prozedur wiederholt zu werden. In diesem Fall wird der Unterzeichner $(L+1)$ als der Verifizierer angesehen; $ID'_0 =$ leere Menge, $X'_0 =$ leere Menge und $y_0 = 0$.

(4-C) Verarbeitung beim Unterzeichner i zur Signaturerzeugung

Schritt S6: Der Unterzeichner i erzeugt die Zufallszahl r_i mittels des Zufallsgenerators 310 und gibt sie in einen n -fach-Punktrechner 320 ein, der die Funktion Φ berechnet, und zwar zusammen mit der öffentlichen Information $\{q, a, b, P\}$, die aus dem Speicher 33 ausgelesen wird, wobei X_i durch die nachstehende Gleichung (60) berechnet wird:

$$X_i = \Phi(r_i, P) = r_i P \text{ über } E_{a,b}(GF(q)) \quad (60).$$

Schritt S7: Der Unterzeichner i verwendet den f_i -Funktionsrechner 330 und den h_i -Funktionsrechner 340 zur Berechnung von e_i bzw. d_i

$$e_i = f_i(X'_i, m'_i) \quad (61)$$

$$d_i = h_i(X'_i, m'_i) \quad (62)$$

wobei

$$X'_i = (X'_{i-1}, X_i) \quad (63)$$

$$m'_i = (m'_{i-1}, m_i) \quad (64).$$

5

Schritt S8: Der Unterzeichner i gibt e_i , d_i und r_i in den Modulo-Multiplizierer 350 und dann in den Modulo-Addierer 360 zusammen mit der öffentlichen Information k und der geheimen Information s_i ein, wobei die Signatur mit der Signaturfunktion Sg_i berechnet wird durch

$$10 \quad y_i = Sg_i(e_i, d_i, s_i, r_i, y_{i-1}) = (y_{i-1} + d_i r_i + e_i s_i) \bmod k \quad (65).$$

Schritt S9: Der Unterzeichner i setzt $ID'_i = (ID'_{i-1}, ID_i)$, und sendet die Nachricht $\{ID'_i, X'_i, m'_i, y_i\}$ an den nächsten Unterzeichner $(i+1)$.

15

(4-D) Verarbeitung durch den Verifizierer zur Signaturverifikation

Fig. 17 zeigt den funktionalen Aufbau der Verifizierervorrichtung 800. Wenn der Verifizierer die Nachricht $\{ID'_L, X'_L, m'_L, y_L\}$ vom Unterzeichner L erhält, verifiziert er die Gültigkeit der einzelnen Signaturen mittels der nachstehend beschriebenen Verarbeitung.

20

Schritt S10: Die ersten i Komponenten der Information X'_L werden zur Bildung von X'_i verwendet, und die ersten i Komponenten der Information m'_L werden zur Bildung von m'_i verwendet. Die Informationen X'_i und m'_i , die auf diese Weise erhalten werden, werden in den f_i -Funktionsrechner 810 und den h_i -Funktionsrechner 820 eingegeben, wo die Komponenten e_i bzw. d_i ($1 \leq i \leq L$) berechnet werden durch

$$25 \quad e_i = f_i(X'_i, m'_i)$$

$$d_i = h_i(X'_i, m'_i).$$

30

Schritt S11: Die Information I_i wird von der ID_i -Komponente in der Information ID'_L abgeleitet, und die Information X_i wird ebenfalls von der Information X'_L abgeleitet. Diese Informationen I_i und X_i werden zusammen mit den oben erwähnten Komponenten e_i und d_i und der aus dem Speicher 88 ausgelesenen öffentlichen Information $\{q, a, b, P\}$ in einen n -fach-Rechner 830 eingegeben, der die Funktion V berechnet, und mit dem Z' berechnet wird durch

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L) = (d_1 X_1 + \dots + d_L X_L + e_1 I_1 + \dots + e_L I_L) \text{ über } E_{a,b}(GF(q)) \quad (66)$$

35

wobei

$$e_i = f_i(X_1, \dots, X_i, \{m_1, \dots, m_i\}) \quad (67)$$

40

$$d_i = h_i(X_1, \dots, X_i, \{m_1, \dots, m_i\}) \quad (68)$$

$(1 \leq i \leq L)$

Schritt S12: Die Information y_L und die öffentliche Information $\{q, a, b, P\}$ werden in den n -fach-Punktrechner 840 eingegeben, der eine Funktion $\Gamma(y_L * P)$ berechnet und damit W gemäß folgender Gleichung errechnet:

45

$$W = \Gamma(y_L, P) = y_L P \text{ über } E_{a,b}(GF(q)) \quad (69).$$

Schritt S13: Z' und W werden in den Komparator 850 eingegeben, wo sie verglichen werden, um sicherzugehen, daß $W = Z'$.

50

Wenn beide übereinstimmen, wird davon ausgegangen, daß die Dokumente (m_1, \dots, m_L) von den L autorisierten Unterzeichnern i jeweils ordnungsgemäß unterzeichnet wurden.

Ausführungsbeispiel 5

55

Dieses Ausführungsbeispiel entspricht dem zweiten Ausführungsbeispiel, welches die Mehrfachsignatur und deren Block-Verifikation durchführt. Nachstehend erfolgt die Beschreibung eines Ausführungsbeispiels, bei dem das Schnorr-Schema auf die Mehrfachsignatur und deren Block-Verifikation angewendet wird, die das Verfahren der elliptischen Kurve einsetzen. Auch bei diesem Ausführungsbeispiel kann die Idee der Verwendung der zweiten Mehrfachkomponente in weitem Umfang auf die ElGamal-Signatur-Schemata und die digitalen Signatur-Schemata, die die interaktiven Prüfungen unter Einschluß derselben verwenden, angewendet werden.

60

Die Systemkonfiguration, auf die dieses Ausführungsbeispiel angewendet wird, ist dieselbe wie die in Fig. 1A gezeigte, und der Aufbau der Zentrale 100 ist der gleiche wie der in Fig. 14 gezeigte.

(5-A) Anfängliche Informationseinstellverarbeitung

65

Es erfolgt nachstehend unter Bezugnahme auf Fig. 14 eine Beschreibung der anfänglichen Informationseinstellverarbeitung in dem Moment, wenn die Zentrale 100 das System startet.

Schritt S1: Die Zentrale 100 erzeugt die Primzahl q mittels des Primzahlgenerators 110 und $a, b, GF(q)$ mittels des Pa-

rametergenerators 120.

Schritt S2: Die Zentrale 100 erzeugt den Punkt $P \in E_{a,b}(GF(q))$ auf der elliptischen Kurve mittels des Basispunktgenerators 130, der eine Funktion $G_1(q)$ berechnet, und die Ordnung k des Basispunkts P mittels des Ordnungsrechners 140. Der Punkt P entspricht dem früher erwähnten Parameter β .

Schritt S3: Die öffentliche Information $\{q, a, b, P, k\}$ wird an die Unterzeichnervorrichtungen 30₁, ..., 30_L und die Verifizierervorrichtung 800 über die sicheren Kommunikationskanäle 400 gesendet und in deren Speichern 33 bzw. 88 gespeichert. 5

Wie schon früher erwähnt, kann der Ordnungsrechner 140 leicht unter Verwendung des Schoof-Algorithmus implementiert werden, der die Ordnung der elliptischen Kurve $E_{a,b}(GF(q))$ (d. h. die Anzahl von Punkten auf der Kurve) berechnet. 10

(5-B) Verarbeitung beim Unterzeichner i , wenn dieser sich in das System einschreibt

Als nächstes erfolgt unter Bezugnahme auf Fig. 18 eine Beschreibung der Verarbeitung, die beim Unterzeichner i ausgeführt wird, wenn er sich in das System einschreibt. 15

Schritt S4: Der Unterzeichner i erzeugt die Zufallszahl s_i mittels des Zufallsgenerators 310 und gibt sie sowie die öffentliche Information $\{q, a, b, P\}$ in den n -fach-Punktrechner 320 ein, der eine Funktion $G_2(s_i, P)$ berechnet und in welchem die öffentliche Information I_i auf folgende Weise errechnet wird

$$I_i = G_2(s_i, P) = s_i P \text{ über } E_{a,b}(GF(q)) \quad (70). \quad 20$$

Schritt S5: Der Unterzeichner i sendet die Identifikationsinformation ID_i , die öffentliche Information I_i und die Einweg-Funktionen f_i und h_i über den sicheren Kommunikationskanal 400 an die Zentrale 100, um sie dort als öffentliche Information $\{ID_i, I_i, f_i, h_i\}$ registrieren zu lassen. Jeder Unterzeichner behält die jeweilige Zufallszahl s_i als geheime Information. 25

In der nachfolgenden Beschreibung wird die unterzeichnete Version des Dokuments m_i , die von dem Unterzeichner i geliefert wird, durch $\{I'_i, X'_i, m_i, y_i\}$ identifiziert. Die Interaktionsfolge der Nachricht ist die gleiche wie im Fall von Fig. 7. Bei Empfang einer Nachricht $\{ID'_{i-1}, X'_{i-1}, m, y_{i-1}\}$ von dem Unterzeichner $(i-1)$ führt der Unterzeichner i die nachstehend beschriebene Signaturerzeugungsverarbeitung durch. Die Konfiguration des Unterzeichners i (der Unterzeichnervorrichtung 30_i) ist in Fig. 18 gezeigt. Es erfolgt nun eine Beschreibung des Falles, wo der Unterzeichner $(i-1)$ die zu unterzeichnende Nachricht sendet und der Unterzeichner i seine Signatur an der Nachricht anbringt und die unterzeichnete Nachricht an den nächsten Unterzeichner $(i+1)$ sendet. Wenn L Unterzeichner eine Mehrfachsignatur erzeugen, braucht lediglich i schrittweise um eins von 1 bis L erhöht zu werden und die folgende Prozedur wiederholt zu werden. In diesem Fall wird der Unterzeichner $(i+1)$ als der Verifizierer betrachtet; $ID'_0 = \text{leere Menge}$, $X'_0 = \text{leere Menge}$ und $y_0 = 0$. 30
35

(5-C) Verarbeitung beim Unterzeichner i zur Signaturerzeugung

Schritt S6: Der Unterzeichner i erzeugt die Zufallszahl r_i mittels des Zufallsgenerators 310 und gibt sie in den n -fach-Punktrechner 320 ein, der die Funktion Φ berechnet, und zwar zusammen mit der öffentlichen Information $\{q, a, b, P\}$, die aus dem Speicher 33 ausgelesen wird, wodurch in nachstehender Weise X_i berechnet wird 40

$$X_i = \Phi(r_i, P) = r_i P \text{ über } E_{a,b}(GF(q)) \quad (71).$$

Schritt S7: Die Unterzeichner i berechnet e_i und d_i unter Verwendung des f_i -Funktionsrechners 330 bzw. des h_i -Funktionsrechners 340 durch 45

$$e_i = f_i(X'_i, m) \quad (72)$$

$$d_i = h_i(X'_i, m) \quad (73) \quad 50$$

wobei $X'_i = (X'_{i-1}, X_i)$.

Schritt S8: Der Unterzeichner i gibt e_i , d_i , r_i und y_{i-1} in den Modulo-Multiplizierer 350 und dann in den Modulo-Addierer 360 zusammen mit der öffentlichen Information k und der geheimen Information s_i ein, wodurch die Signatur mit der Signaturfunktion Sg_i durch Gleichung (74) erzeugt wird: 55

$$y_i = Sg_i(e_i, d_i, s_i, r_i, y_{i-1}) = (y_{i-1} + d_i r_i + e_i s_i) \bmod k \quad (74).$$

Schritt S9: Der Unterzeichner i setzt $ID'_i = (ID'_{i-1}, ID_i)$, und sendet die Nachricht $\{ID'_i, X'_i, m, y_i\}$ an den nächsten Unterzeichner $(i+1)$. 60

(5-D) Verarbeitung beim Verifizierer zur Signaturverifikation

Fig. 19 zeigt den funktionalen Aufbau der Verifizierervorrichtung 800. Wenn der Verifizierer die Nachricht $\{ID'_L, X'_L, m, y_L\}$ von dem Unterzeichner L empfängt, verifiziert er die Gültigkeit der einzelnen Signaturen mittels der nachfolgend beschriebenen Verarbeitung. 65

Schritt S10: Die ersten i Komponenten der Information X'_L werden zur Bildung von X'_i verwendet, das zusammen mit der Nachricht m in den f_i -Funktionsrechner 810 und den h_i -Funktionsrechner 820 eingegeben wird, worin die Kompo-

nenten e_i bzw. d_i ($1 \leq i \leq L$) berechnet werden durch

$$e_i = f_i(X'_i, m)$$

$$5 \quad d_i = h_i(X'_i, m).$$

Schritt S11: Die Information I_i wird von der ID_i -Komponente in der Information ID'_L abgeleitet, und außerdem wird die Information X_i von der Information X'_L abgeleitet. Diese Informationen I_i und X_i werden zusammen mit den oben erwähnten Komponenten e_i und d_i sowie der öffentlichen Information $\{q, a, b, P, k\}$, die aus dem Speicher 88 ausgelesen wird, in den n -fach-Rechner 830 eingegeben, der die Funktion V berechnet, und von dem Z' errechnet wird durch

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L) = (d_1 X_1 + \dots + d_L X_L + e_1 I_1 + \dots + e_L I_L) \text{ über } E_{a,b}(GF(q)) \quad (75)$$

wobei

$$15 \quad e_i = f_i(X_1, \dots, X_i, m) \quad (76)$$

$$d_i = h_i(X_1, \dots, X_i, m) \quad (77)$$

$$(1 \leq i \leq L)$$

20 Schritt S12: Die Information y_L und die öffentliche Information $\{q, a, b, P, k\}$ werden in den n -fach-Punktrechner 840 eingegeben, der die Funktion $\Gamma(y_L * P)$ berechnet und dadurch W wie folgt errechnet:

$$W = \Gamma(y_L, *P) = y_L P \text{ über } E_{a,b}(GF(q)) \quad (78).$$

25 Schritt S13: Z' und W werden in den Komparator 850 eingegeben, wo sie miteinander verglichen werden, um sicherzugehen, daß $W = Z'$.

Wenn beide übereinstimmen, wird davon ausgegangen, daß das Dokument in von den L autorisierten Unterzeichnern ordnungsgemäß unterzeichnet wurde.

30 Jede Vorrichtung kann so ausgebildet werden, daß sie ihre Funktionen durch Lesen, Übersetzen und Ausführen von Programmen unter Verwendung eines Computers durchführt. Bei jeder Unterzeichnervorrichtung kann $ID'_i = (ID'_{i-1}, ID_i)$ ersetzt werden durch $ID'_i = (ID'_{i-1}, I_i)$. Dies erfordert die Speicherung der öffentlichen Information I_i in Speichermitteln und entlastet damit die Verifizierungsvorrichtung, die dann nicht mehr die Information I_i in der Identifikationsinformation ID_i lokalisieren muß.

Ausführungsbeispiel 6

Dieses Ausführungsbeispiel entspricht dem dritten Ausführungsbeispiel, bei dem mehrere Unterzeichner einzeln ihre Signatur an einem jeweiligen Dokument anbringen und die Signaturen en-bloc verifiziert werden. Dieses Ausführungsbeispiel wird ebenfalls in Verbindung mit dem Fall der Anwendung des Schnorr-Schemas und des Einsatzes des Verfahrens der elliptischen Kurve beschrieben. Auch bei diesem Ausführungsbeispiel kann die Idee der Verwendung der zweiten Mehrfachkomponente in weitem Umfang auf die ElGamal-Signatur-Schemata und die digitalen Signatur-Schemata angewendet werden, die die interaktiven Prüfungen einsetzen, welche dieselben enthalten.

Die Systemkonfiguration, auf die dieses Ausführungsbeispiel angewendet wird, ist die gleiche wie die in Fig. 1B gezeigte, und die Konfiguration der Zentrale 100 ist die gleiche wie die in Fig. 14 gezeigte.

45 Bei der folgenden Beschreibung wird die unterzeichnete Nachricht durch $\{ID_i, X_i, m_i, y_i\}$ repräsentiert, und zwar unter der Annahme, daß der Unterzeichner i das Dokument m_i unterzeichnet.

Die Interaktionsfolge der Nachricht ist die gleiche wie die in Fig. 11 gezeigte. Fig. 20 zeigt in Blockform die Unterzeichnervorrichtung 30_i.

(6-A) Verarbeitung beim Unterzeichner i zur Signaturerzeugung

Schritt S14: Der Unterzeichner i erzeugt die Zufallszahl r_i mittels des Zufallsgenerators 310 und gibt sie zusammen mit der öffentlichen Information $\{q, a, b, P, k\}$, die aus dem Speicher 33 ausgelesen wird, in den n -fach-Punktrechner 320 ein, der die Funktion Φ berechnet, und in dem X_i berechnet wird durch

$$X_i = \Phi(r_i, P) = r_i P \text{ über } E_{a,b}(GF(q)) \quad (79).$$

60 Schritt S15: Der Unterzeichner i berechnet e_i und d_i unter Verwendung des f_i -Funktionsrechners 330 und des h_i -Funktionsrechners 340 mittels

$$e_i = f_i(X_i, m) \quad (80)$$

$$d_i = h_i(X_i, m) \quad (81).$$

65 Schritt S16: Der Unterzeichner i gibt e_i , d_i und r_i in den Modulo-Multiplizierer 350 und dann den Modulo-Addierer 360 zusammen mit der öffentlichen Information k und der geheimen Information s_i ein, wodurch die Signatur mit der Signaturfunktion Sg_i erzeugt wird durch

$$y_i = S_{g_i}(c_i, d_i, s_i, r_i, k) = (d_i r_i + c_i s_i) \bmod k \quad (82).$$

Schritt S17: Der Unterzeichner i sendet die Nachricht $\{ID_i, X_i, m, y_i\}$ an den Verifizierer **800**.

(6-B) Verarbeitung beim Verifizierer zur Signaturverifikation

Fig. 21 zeigt in Blockform die Verifizierervorrichtung **800**. Wenn der Verifizierer L Nachrichten $\{I_i, X_i, m_i, y_i\}$ von den L Unterzeichnern empfängt verifiziert er die Gültigkeit der einzelnen Signaturen durch die nachstehende Verarbeitung.

Schritt S18: Der Verifizierer **800** gibt die Information X_i und die Nachricht m_i in den f_i -Funktionsrechner **810** und den h_i -Funktionsrechner **820** ein, worin die Komponenten e_i bzw. d_i ($1 \leq i \leq L$) errechnet werden durch

$$e_i = f_i(X_i, m)$$

$$d_i = h_i(X_i, m_i).$$

Schritt S19: Der Verifizierer **800** leitet die Information I_i von der ID_i -Komponente und die Information X_i von der Information X_L ab und gibt sie zusammen mit den oben erwähnten Komponenten e_i und d_i sowie der öffentlichen Information $\{q, a, b, P, k\}$, die aus dem Speicher **88** ausgelesen wird, in den n -fach-Rechner **830** ein, der die Funktion V berechnet und in dem Z' errechnet wird durch

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L) = (d_1 X_1 + \dots + d_L X_L + e_1 I_1 + \dots + e_L I_L) \text{ über } E_{a,b}(GF(q)) \quad (83)$$

wobei

$$e_i = f_i(X_1, \dots, X_i, \{m_1, \dots, m_i\}) \quad (84)$$

$$d_i = h_i(X_1, \dots, X_i, \{m_1, \dots, m_i\}) \quad (85)$$

$$(1 \leq i \leq L)$$

Schritt S20: Der Verifizierer **800** gibt die L Informationen y_i und die öffentliche Information k in den Modulo-Addierer **840** ein, in welchem ein akkumulierter Wert Y berechnet wird durch

$$Y = \sum_{i=1}^L y_i \bmod k \quad (86)$$

und gibt diesen in den n -fach-Punktrechner **845** ein, der die Funktion $\Gamma(Y * P)$ und so weiter berechnet und dadurch W in folgender Weise errechnet:

$$W = \Gamma(Y * P) = YP \text{ über } E_{a,b}(GF(q)) \quad (87).$$

Schritt S21: Z' und W werden in den Komparator **850** eingegeben, wo sie miteinander verglichen werden um sicherzustellen, daß $W = Z'$.

Wenn beide miteinander übereinstimmen, wird davon ausgegangen, daß die L Dokumente m_i jeweils ordnungsgemäß von den L autorisierten Unterzeichnern i unterzeichnet wurden.

Unter Berücksichtigung der Art der Erzeugung des akkumulierten Werts Y gilt

$$Y P \equiv (y_{L-1} P) + \{d_L(r_L P)\} + \{e_L(s_L P)\} \equiv y_L P + d_L X_L + e_L I_L = \dots \equiv (d_1 X_1 + \dots + d_L X_L + e_1 I_1 + \dots + e_L I_L) \text{ über } E_{a,b}(GF(q)) \quad (88).$$

Wenn also der obige Vergleichstest durch den Komparator **850** bestanden wird, akzeptiert der Verifizierer **800** die Dokumente in $(i = 1, \dots, L)$ als durch L autorisierte Unterzeichner jeweils ordnungsgemäß unterzeichnet.

Bewertung der Ausführungsbeispiele

Es erfolgt nun eine Bewertung der vorliegenden Erfindung im Vergleich mit dem RSA-Schema und dem Schnorr-Schema hinsichtlich der Rechenkomplexität grundlegender Operationen, die in den Unterzeichnungs- und Verifikations-Prozeduren enthalten sind, sowie der Redundanz verwendeter Nachrichten und in anderer Hinsicht. Das zu bewertende System ist eines, welches die Mehrfachsignatur und deren Verifikation ausführt. Dementsprechend werden das zweite und das fünfte Ausführungsbeispiel der vorliegenden Erfindung bewertet.

Die Tabelle der Fig. 22 zeigt im Vergleich grundlegende Berechnungen für die Mehrfachsignatur und ihre Verifikation in dem RSA-Schema, dem Schnorr-Schema sowie dem zweiten und dem fünften Ausführungsbeispiel der Erfindung. Die Tabelle von Fig. 23 zeigt die Anzahl von in den Unterzeichnungs- und Verifikations-Prozeduren der jeweiligen Schemata enthaltenen Berechnungen, wenn die Gleichungen in Fig. 22 verwendet werden. Fig. 23 zeigt außerdem die Redundanz von Nachrichten, die Anzahl von Kommunikationen, die zur Verifizierung aller Signaturen erforderlich sind und die Anzahl von Zirkulationsdurchläufen jeder Nachricht.

(1) Rechenumfang beim Unterzeichner

Bei den Operationen zum Zwecke der Unterzeichnung, die in der Tabelle von Fig. 22 gezeigt sind, sind die Berechnungen der Einweg-Funktionen f , f_i und h_i schneller als die Multiplikation und die n -fach-Punktrechnung. Daher wird der Rechenaufwand jeder Unterzeichnervorrichtung im Hinblick auf die Anzahl von Modulo- N -Multiplikationen (einschließlich Modulo- N - oder - p -Rechnungen) und die Anzahl von Berechnungen verglichen, die für den n -fach-Punkt auf der elliptischen Kurve durchgeführt werden.

Gewöhnlich wird $|N|=1024$ und $|q|=160$ empfohlen. In diesem Fall ist, da die führenden Terme jeweils der ersten Berechnung entsprechen, die Verarbeitungsgeschwindigkeit bei dem zweiten und dem fünften Ausführungsbeispiel mehr als fünfmal so groß wie die im Fall unter Verwendung des RSA-Verschlüsselungssystems, wie in Fig. 23 gezeigt. Es wurde berichtet, daß die Berechnung des n -fach-Punkts auf der elliptischen Kurve, wie bei dem fünften Ausführungsbeispiel, etwa zehnmal so schnell wie die Unterzeichnungsprozedur unter Verwendung des RSA-Verschlüsselungssystems ist (siehe beispielsweise <http://www.certicom.com/html/eccqa.html>).

(2) Rechenumfang beim Verifizierer

Bei den Berechnungen zur Signaturverifikation, die in Fig. 22 gezeigt sind, sind die Berechnungen der Einweg-Funktionen f , f_i und h_i schneller als die Multiplikation und die Berechnung des n -fach-Punkts auf der elliptischen Kurve. Daher wird der Verarbeitungsumfang beim Verifizierer im Hinblick auf die Anzahl von Potenzierungen (einschließlich Modulo- N - oder - p -Berechnungen) und die Anzahl von Berechnungen für den n -fach-Punkt auf der elliptischen Kurve verglichen. Wie in Fig. 23 gezeigt, ist die Rechenbelastung für die Signaturverifikation durch die vorliegende Erfindung die gleiche wie im Fall der Verwendung des RSA-Verschlüsselungssystems, aber im wesentlichen halb so groß wie diejenigen im Fall der Verwendung des Schnorr-Schemas.

(3) Nachrichtenredundanz

Bei jedem Multi-Signaturschema wird die ID-Information der Nachricht für jede einzelne Signatur (die ID_L -Komponenten) im Hinblick darauf hinzugefügt, daß der Unterzeichner deutlich wird. Im folgenden wird die Redundanz der Nachricht $\{ID, X, m, y\}$ unter Verwendung der Anzahl von Bits jeweils der X - und der y -Komponente bewertet. Die Signaturkomponente (y -Komponente) bei Verwendung des RSA-Verschlüsselungssystems ist repräsentiert durch $D_L \dots D_1(f(m))$. Die Ergebnisse sind in Fig. 23 gezeigt.

Bei dem Verfahren des zweiten und des fünften Ausführungsbeispiels der vorliegenden Erfindung ergibt sich $L \times |X| + |y|$ Bits. Die Anwendung des fünften Ausführungsbeispiels liefert $|p| = |q| = |e| = 160$ und $|X| = |q| + 1$; d. h. $161L + 160$ Bits. Dies zeigt an, daß für den Fall von $2 \leq L \leq 6$ das Verfahren des fünften Ausführungsbeispiels vorteilhaft ist.

(4) Anzahl von Kommunikationen und Anzahl von Zirkulationsdurchläufen

Wie zuvor in Verbindung mit dem Hintergrund der vorliegenden Erfindung erläutert, beinhalten die Multi-Signaturprozedur und die Verifikationsprozedur mit dem Schnorr-Schema zwei Zirkulationsrunden oder Umläufe der Nachricht zu den Unterzeichnern. Daraus ergibt sich, daß die erforderliche Anzahl von Kommunikationen ebenfalls doppelt so groß wie die bei den anderen Schemata ist.

Was die Basis der Sicherheit des Signaturschemas und des Block-Verifikationsschemas gemäß der vorliegenden Erfindung angeht, schließt die Schwierigkeit des diskreten Logarithmusproblems durch das Modulo- p jede Erfolgsmöglichkeit zur Berechnung der geheimen Information s_i aus der öffentlichen Information $\{p, q, g, I_i\}$ aus. Daß jede Unterzeichnervorrichtung nicht die Mehrfachsignatur einschließlich der Signatur irgendeiner anderen Unterzeichnervorrichtung fälschen kann, kann garantiert werden durch Kombination der Verfahren der vorliegenden Erfindung mit der "Exakte Sicherheit"-Eigenschaft unter dem sogenannten "Random-Oracle-Modell", das Ergebnis der theoretischen Studien der Rechenkomplexität ist.

Hinsichtlich der "Exakte Sicherheit"-Eigenschaft wird beispielsweise verwiesen auf die Fundstelle M. Bellare and P. Rogaway, "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols", Proc. of the First ACM Conference on Computer and Communications Security, Seiten 62-73, und K. Ohta and T. Okamoto, "The Exact Security of Multi-Signature Schemes", Technical Report of IEICE ISEC97-27.

Wie oben beschrieben, kann gemäß der vorliegenden Erfindung die Signaturerzeugungsverarbeitung mehr als fünfmal so schnell wie im Fall der Verwendung des RSA-Schemas ausgeführt werden. Die Signaturverifikationsverarbeitung stimmt in der Geschwindigkeit mit dem Fall der Verwendung des RSA-Schemas überein, kann jedoch doppelt so schnell gemacht werden wie im Fall der wiederholten Verwendung des Schnorr-Schemas. Wenn L gleich 6 oder kleiner ist, ist die Redundanz der Nachricht bei der vorliegenden Erfindung die gleiche wie im Fall der wiederholten Verwendung des Schnorr-Schemas und ist vorteilhafter als im Fall der Verwendung des RSA-Schemas.

Patentansprüche

1. Verfahren, mit dem ein Verifizierer en-bloc digitale Signaturen verifiziert, die von einer Reihe von Unterzeichnern i , $i = 1, 2, \dots, L$, wobei L eine ganze Zahl gleich oder größer als zwei ist, an einem in elektronischer Form vorliegenden Dokument m_i angebracht wurden, wobei Information enthaltend einen Parameter q für jeden der Unterzeichner i zur Erzeugung einer Signaturfunktion S_i und einen Parameter $\beta = G_1(q)$, der mit einer Funktion G_1 unter Verwendung des Parameters q erhalten wird, vorab veröffentlicht wird, umfassend die Schritte:

– jeder Unterzeichner i

(a) erzeugt eine erste Zufallszahl s_i als geheime Information, erzeugt dann Information $I_i = G_2(s_i, \beta)$ mit

einer Funktion G_2 unter Verwendung des öffentlichen Parameters β und der ersten Zufallszahl s_i und veröffentlicht die Information I_i sowie zwei Einweg-Funktionen f_i und h_i und Identifikationsinformation ID_i , die von dem Unterzeichner i verwendet wird, als seine öffentliche Information $\{ID_i, I_i, f_i, h_i\}$;

(b) erzeugt eine zweite Zufallszahl r_i , erzeugt dann Information $X_i = \Phi(r_i, \beta)$, indem der Parameter β und die zweite Zufallszahl r_i in eine Funktion Φ eingesetzt werden, und setzt Information, die die Information X_i enthält, auf X'_i ;

(c) erzeugt

$$e_i = f_i(X'_i, m'_i)$$

$$d_i = h_i(X'_i, m'_i)$$

mit den Einweg-Funktionen f_i und h_i unter Verwendung von Dokumentinformation m'_i , die ein zu unterzeichnendes Dokument m_i enthält, sowie der Information X'_i ; und

(d) erzeugt für Information enthaltend e_i, d_i, s_i und r_i eine Signatur

$$y_i = Sg_i(e_i, d_i, s_i, r_i, y_{i-1})$$

mit der Signaturfunktion Sg_i , die unter Verwendung des Parameters q erzeugt wird, und sendet Information $\{ID'_i, X'_i, m'_i, y_i\}$, die die Identifikationsinformation ID_i als Identifikationsinformation ID'_i enthält, an den nächsten Unterzeichner $(i+1)$ in der Reihe von Unterzeichnern $i = 1$ bis L , wobei der letzte Unterzeichner L die Information $\{ID'_L, X'_L, m'_L, y_L\}$ an den Verifizierer als letzte Bestimmung sendet; und

– der Verifizierer

(e) berechnet, aus der öffentlichen Information $\{ID_i, I_i, f_i, h_i\}$, Information I_i entsprechend der Identifikationsinformation ID_i , die in der Information ID'_i in der empfangenen Information $\{ID'_L, X'_L, m'_L, y_L\}$ enthalten ist, und die Einweg-Funktionen f_i und h_i und berechnet e_i und d_i unter Verwendung der Einweg-Funktionen f_i und h_i sowie X'_i und m'_i in den empfangenen Informationen X'_L und m'_L ;

(f) berechnet die Information X_i , die in der Information X'_i enthalten ist, und berechnet

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L)$$

mit einer Funktion V , enthaltend Berechnungen von $(X_i * d_i)$ aus d_i und X_i und $(I_i * e_i)$ aus e_i und I_i , für $i = 1, \dots, L$; und

(g) berechnet $W = \Gamma(y_L * \beta)$ mit einer Funktion Γ , die eine Berechnung von $(y_L * \beta)$ von y_L und β enthält, prüft dann ob $W = Z'$ und entscheidet, falls beide Werte miteinander übereinstimmen, daß die Signaturen alle gültig sind.

2. Verfahren, bei dem ein Verifizierer en-bloc digitale Signaturen verifiziert, die von Unterzeichnern $i, i = 1, 2, \dots, L$, wobei L eine ganze Zahl gleich oder größer als zwei ist, an einem in elektronischer Form vorliegenden Dokument m'_i angebracht wurden, wobei Information enthaltend einen Parameter q für jeden der Unterzeichner i zur Erzeugung einer Signaturfunktion Sg_i und einen Parameter $\beta = G_1(q)$, der unter Verwendung des Parameters q mit einer Funktion G_1 erhalten wird, vorab veröffentlicht wird, umfassend die Schritte:

– jeder Unterzeichner i

(a) erzeugt eine erste Zufallszahl s_i als geheime Information, erzeugt dann Information $I_i = G_2(s_i, \beta)$ mit einer Funktion G_2 unter Verwendung des öffentlichen Parameters β und der ersten Zufallszahl s_i und veröffentlicht die Information I_i sowie zwei Einweg-Funktionen f_i und h_i und Identifikationsinformation ID_i , die von dem Unterzeichner i verwendet wird, als seine öffentliche Information $\{ID_i, I_i, f_i, h_i\}$;

(b) erzeugt eine zweite Zufallszahl r_i , erzeugt dann Information $X_i = \Phi(r_i, \beta)$, indem der Parameter β und die zweite Zufallszahl r_i in eine Funktion Φ eingesetzt werden, und setzt Information, die die Information X_i enthält, auf X'_i ;

(c) erzeugt

$$e_i = f_i(X'_i, m'_i)$$

$$d_i = h_i(X'_i, m'_i)$$

mit den Einweg-Funktionen f_i und h_i unter Verwendung von Dokumentinformation, die ein zu unterzeichnendes Dokument in enthält, sowie der Information X'_i ; und

(d) erzeugt für Information enthaltend e_i, d_i, s_i und r_i eine Signatur

$$y_i = Sg_i(e_i, d_i, s_i, r_i)$$

mit der Signaturfunktion Sg_i , die unter Verwendung des Parameters q erzeugt wird, und sendet die Information $\{ID'_i, X'_i, m'_i, y_i\}$, die die Identifikationsinformation ID_i als Identifikationsinformation ID'_i enthält, an den Verifizierer als letzte Bestimmung; und

– der Verifizierer

(e) berechnet aus der öffentlichen Information $\{ID_i, I_i, f_i, h_i\}$ Information I_i entsprechend der Identifikationsinformation ID_i , die in der Information ID'_i in der empfangenen Information $\{ID'_i, X'_i, m'_i, y_i\}$ enthalten ist, und die Einweg-Funktionen f_i und h_i und berechnet e_i und d_i unter Verwendung der Einweg-

Funktionen f_i und h_i und der empfangenen Informationen X'_i und m'_i ;
 (f) berechnet die Information X_i , die in der Information X'_i enthalten ist, und berechnet

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L)$$

mit einer Funktion V , die Berechnungen von $(X_i * d_i)$ aus d_i und X_i sowie von $(I_i * e_i)$ aus e_i und I_i enthält, für $i = 1, \dots, L$; und

(g) berechnet $W = \Gamma(Y * \beta)$ mit einer Funktion Γ , die eine Berechnung von $(Y * \beta)$ unter Verwendung von β und eines akkumulierten Wert Y von y_1 bis y_L enthält, prüft dann, ob $W = Z'$ und entscheidet, wenn beide Werte miteinander übereinstimmen, daß die Signaturen alle gültig sind.

3. Verfahren nach Anspruch 1, bei dem:

$$X'_i = (X'_{i-1}, X_i),$$

$$m'_i = (m'_{i-1}, m_i),$$

$$ID'_i = (ID'_{i-1}, ID_i),$$

$$X_0 = \text{leere Menge}$$

$$m'_0 = \text{leere Menge}$$

$$ID'_0 = \text{leere Menge}$$

$$y_0 = 0; \text{ und}$$

der Unterzeichner i von dem Unterzeichner $(i-1)$ $(ID'_{i-1}, X'_{i-1}, m'_{i-1}, y_{i-1})$ empfängt und die Schritte (b) bis (d) ausführt, und Information $\{ID'_i, X'_i, y_i\}$ an den Unterzeichner $(i+1)$ sendet, während der letzte Unterzeichner L die Schritte (b) bis (d) mit von dem Unterzeichner $(i-1)$ erhaltender Information durchläuft und Information $\{ID'_L, X'_L, m'_L, y_L\}$ erzeugt und an den Verifizierer sendet.

4. Verfahren nach Anspruch 3, bei dem $m'_1 = m_1 = m$; $m_2 = m_3 = \dots = m_L = \text{leere Menge}$; und

der Unterzeichner i von dem Unterzeichner $(i-1)$ Information $\{ID'_{i-1}, X'_{i-1}, m, y_{i-1}\}$ empfängt und

die Schritte (b) bis (d) ausführt und Information $\{ID'_i, X'_i, m, y_i\}$ an den Unterzeichner $(i+1)$ sendet, während der letzte Unterzeichner L die Schritte (b) bis (d) mit von dem Unterzeichner $(i-1)$ erhaltener Information ausführt, Information $\{ID'_L, X'_L, m, y_L\}$ erzeugt und diese an den Verifizierer sendet.

5. Verfahren nach Anspruch 2, bei dem $X'_i = X_i$, $m'_i = m_i$, und $ID'_i = ID_i$; und jeder Unterzeichner die Schritte (b) bis (d) ausführt und Information $\{ID'_i, X'_i, m_i, y_i\}$ erzeugt und einzeln an den Verifizierer sendet.

6. Verfahren nach Anspruch 3 oder 4, bei dem $ID'_i = (ID'_{i-1}, ID_i)$ ersetzt wird durch $ID'_i = (ID'_{i-1}, I_i)$.

7. Verfahren nach Anspruch 3 oder 4, bei dem, wenn die Anzahl von Elementen einer Gruppe p ist, ein Element g der Gruppe, bei dem eine Gruppenrechnung beginnt, der Parameter β ist und eine ganze Zahl, mit der, wenn das Element g q -mal gruppenberechnet wird, die Berechnung zu g zurückkehrt, der Parameter q ist, und diese Parameter $\{p, q, g\}$ als öffentliche Systeminformation veröffentlicht werden,

die Berechnung der Information I_i unter Verwendung der Funktion $G_2(s_i, g)$ in Schritt (a) durch eine s_i -malige Gruppenberechnung des Parameters g unter Verwendung des Parameters p durchgeführt wird,

die Berechnung der Information X_i unter Verwendung der Funktion Φ im Schritt (b) durch r_i -malige Gruppenberechnungen des Parameters g unter Verwendung des Parameters p durchgeführt wird,

der Schritt (f) ein Schritt zum Erhalt von Z' durch sequentielle Berechnung von Werten $(X_i * d_i)$, erhalten durch d_i -malige Multikomponenten-Gruppenberechnungen von X_i , und Werten $(I_i * e_i)$, erhalten durch e_i -malige Multikomponenten-Gruppenberechnungen von I_i , für jeden Unterzeichner i von 1 bis L ist, und

der Schritt (g) ein Schritt der Berechnung von W mittels y_L -maligen Berechnungen des Parameters g unter Verwendung der empfangenen Information y_L und der öffentlichen Informationen p und q ist.

8. Verfahren nach Anspruch 7, ferner umfassend einen Schritt der Vorabberzeugung von p und q , bei denen es sich um Primzahlen handelt, für die die Beziehung gilt $1 = p \bmod q$, sowie des Erzeugens eines Grundelements α von $(Z/pZ)^*$, und bei dem die Funktion G_1 zur Berechnung des Parameters $\beta = g$ durch die folgende Gleichung gegeben ist:

$$g = G_1(q) = \alpha^{(p-1)/q} \bmod p;$$

die Funktion $G_2(s_i, g)$ im Schritt (a) durch die folgende Gleichung gegeben ist:

$$I_i = G_2(s_i, g) = g^{s_i} \bmod p,$$

die Funktion Φ in Schritt (b) durch die folgende Gleichung gegeben ist:

$$X_i = \Phi(r_i, g) = g^{r_i} \bmod p,$$

die Signaturfunktion Sg_i in Schritt (d) gegeben ist durch die folgende Gleichung unter Verwendung von e_i, d_i, r_i, s_i, q und y_{i-1} :

$$y_i = Sg_i(e_i, d_i, s_i, r_i, y_{i-1}) = (y_{i-1} + d_i r_i + e_i s_i) \bmod q,$$

die Funktion V im Schritt (f) gegeben ist durch die folgende Gleichung:

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L) = X_1^d I_1^{e_1} \dots X_L^d I_L^{e_L} \bmod p; \text{ und}$$

die Funktion F in Schritt (g) gegeben ist durch die folgende Gleichung:

$$W = \Gamma(y_L * g) = g^Y \text{ mod } p.$$

9. Verfahren nach Anspruch 5, bei dem, wenn die Anzahl von Elementen einer Gruppe p ist, ein Element g der Gruppe, bei dem eine Gruppenrechnung beginnt, der Parameter β ist und eine ganze Zahl mit der, wenn das Element g q -mal gruppenberechnet wird, die Berechnung zu g zurückkehrt, der Parameter q ist, und diese Parameter $\{p, q, g\}$ als öffentliche Systeminformation veröffentlicht werden, 5
die Berechnung der Information I_i unter Verwendung der Funktion $G_2(s_i, g)$ in Schritt (a) durch eine s_i -malige Gruppenberechnung des Parameters g unter Verwendung des Parameters p durchgeführt wird,
die Berechnung der Information X_i unter Verwendung der Funktion Φ im Schritt (b) durch r_i -malige Gruppenberechnungen des Parameters g unter Verwendung des Parameters p durchgeführt wird, 10
der Schritt (f) ein Schritt zum Erhalt von Z' durch sequentielle Berechnung von Werten $(X_i * d_i)$ erhalten durch d_i -malige Multikomponenten-Gruppenberechnungen von X_i und Werten $(I_i * e_i)$, erhalten durch e_i -malige Multikomponenten-Gruppenberechnungen von I_i , für jeden Unterzeichner i von 1 bis L ist, und
der Schritt (g) ein Schritt der Berechnung eines akkumulierten Werts Y unter Verwendung von L Informationen y_i und der Erzeugung, als W , eines Werts (g^Y) ist, der erhalten wird durch Y -maliges Operieren oder Berechnen des Parameters g unter Verwendung von Y und den öffentlichen Informationen p und q . 15
10. Verfahren nach Anspruch 9 ferner umfassend einen Schritt der Vorabzeugung von p und q , bei denen es sich um Primzahlen handelt, für die die Beziehung gilt $1 = p \text{ mod } q$, sowie des Erzeugens eines Grundelements α von $(\mathbb{Z}/p\mathbb{Z})^*$, und bei dem die Funktion G_1 zur Berechnung des Parameters $\beta = g$ durch die folgende Gleichung gegeben ist: 20

$$g = G_1(q) = \alpha^{(p-1)q} \text{ mod } p;$$

die Funktion $G_2(s_i, g)$ im Schritt (a) durch die folgende Gleichung gegeben ist: 25

$$I_i = G_2(s_i, g) = g^{s_i} \text{ mod } p,$$

die Funktion Φ in Schritt (b) durch die folgende Gleichung gegeben ist: 30

$$X_i = \Phi(r_i, g) = g^{r_i} \text{ mod } p,$$

die Signaturfunktion Sg_i in Schritt (d) durch die folgende Gleichung unter Verwendung von e_i, d_i, r_i, s_i , und q gegeben ist: 35

$$y_i = Sg_i(e_i, d_i, s_i, r_i) = (d_i r_i + e_i s_i) \text{ mod } q,$$

die Funktion V im Schritt (f) gegeben ist durch die folgende Gleichung:

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L) = X_1^d I_1^e \dots X_L^d I_L^e \text{ mod } p; \text{ und} \quad 40$$

im Schritt (g) der akkumulierte Wert Y berechnet wird durch

$$Y = \sum_{i=1}^L y_i \text{ mod } q \quad 45$$

und die Funktion Γ in Schritt (g) gegeben ist durch die folgende Gleichung:

$$W = \Gamma(Y * g) = g^Y \text{ mod } p. \quad 50$$

11. Verfahren nach Anspruch 3 oder 4, bei dem der Parameter q ein Parameter eines Definitionsfeldes $GF(q)$ einer elliptischen Kurve $E_{a,b}(GF(q))$ ist, und, wenn ein Basispunkt einer Ordnung k auf der elliptischen Kurve durch den Parameter β repräsentiert ist und ein Parameter der elliptischen Kurve durch $a, b \in GF(q)$, diese Parameter $\{q, a, b, P, k\}$ als öffentliche Systeminformation veröffentlicht werden, und bei dem 55
die Berechnung der Information I_i unter Verwendung der Funktion $G_2(s_i, P)$ im Schritt (a) durch s_i -malige Gruppenberechnungen des Parameters P ausgeführt wird,
die Berechnung der Information X_i unter Verwendung der Funktion Φ in Schritt (b) durch r_i -malige Gruppenberechnungen des Parameters P ausgeführt wird,
der Schritt (f) ein Schritt zum Erhalt von Z' ist durch sequentielles Berechnen von Werten $(X_i * d_i)$, erhalten durch d_i -malige Multikomponenten-Gruppenberechnungen von X_i , und Werten $(I_i * e_i)$, erhalten durch e_i -malige Multikomponenten-Gruppenberechnungen von I_i , für jeden Unterzeichner i von 1 bis L , und 60
der Schritt (g) ein Schritt der Berechnung von W durch y_L -malige Berechnungen des Parameters P auf der elliptischen Kurve unter Verwendung der empfangenen Information y_L und der öffentlichen Information p ist.
12. Verfahren nach Anspruch 11, bei dem die Funktion G_1 zur Berechnung des Parameters $\beta = P$ gegeben ist durch folgende Gleichung: 65

$$G_1(q) = P \in E_{a,b}(GF(q)),$$

die Funktion $G_2(s_i, P)$ in Schritt (a) gegeben ist durch die folgende Gleichung:

$$I_i = G_2(s_i, P) = s_i P \text{ über } E_{a,b}(GF(q));$$

die Funktion Φ in Schritt (b) gegeben ist durch die folgende Gleichung:

$$X_i = \Phi(r_i, P) = r_i P \text{ über } E_{a,b}(GF(q));$$

die Signaturfunktion Sg_i in Schritt (d) gegeben ist durch die folgende Gleichung unter Verwendung von e_i, d_i, e_i, s_i und y_{i-1} :

$$y_i = Sg_i(e_i, d_i, s_i, r_i, y_{i-1}) = (y_{i-1} + d_i r_i + e_i s_i) \bmod k,$$

die Funktion V im Schritt (f) gegeben ist durch die folgende Gleichung:

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L) = (d_1 X_1 + \dots + d_L X_L + e_1 I_1 + \dots + e_L I_L) \text{ über } E_{a,b}(GF(q)); \text{ und}$$

die Funktion Γ im Schritt (g) gegeben ist durch die folgende Gleichung:

$$W = \Gamma(y_L * P) = y_L P \text{ über } E_{a,b}(GF(q)).$$

13. Verfahren nach Anspruch 5, bei dem der Parameter q ein Parameter eines Definitionsfeldes $GF(q)$ einer elliptischen Kurve $E_{a,b}(GF(q))$ ist, und, wenn ein Basispunkt einer Ordnung k auf der elliptischen Kurve repräsentiert ist durch den Parameter β , ein Parameter der elliptischen Kurve durch $a, b \in GF(q)$, diese Parameter $\{q, a, b, P, k\}$ als öffentliche Systeminformation veröffentlicht wird, und bei dem die Berechnung der Information I_i unter Verwendung der Funktion $G_2(s_i, P)$ in Schritt (a) durchgeführt wird durch s_i -malige Gruppenberechnungen des Parameters P , die Berechnung der Information X_i unter Verwendung der Funktion Φ in Schritt (b) durchgeführt wird durch r_i -malige Gruppenberechnungen des Parameters P , Schritt (f) ein Schritt zum Erhalt von Z' durch sequentielle Berechnung von Werten $(X_i * d_i)$, erhalten durch d_i -malige Multikomponenten-Gruppenberechnungen von X_i , und Werten $(I_i * e_i)$, erhalten durch e_i -malige Multikomponenten-Gruppenberechnungen von I_i , für jeden Unterzeichner i von 1 bis L , und der Schritt (g) ein Schritt der Berechnung von W durch Y -malige Berechnungen des Parameters P auf der elliptischen Kurve unter Verwendung der empfangenen Information y_L und der öffentlichen Information p ist.

14. Verfahren nach Anspruch 13, bei dem die Funktion G_1 zur Berechnung des Parameters $\beta = P$ gegeben ist durch folgende Gleichung:

$$G_1(q) = P \in E_{a,b}(GF(q)),$$

die Funktion $G_2(s_i, P)$ in Schritt (a) gegeben ist durch die folgende Gleichung:

$$I_i = G_2(s_i, P) = s_i P \text{ über } E_{a,b}(GF(q));$$

die Funktion Φ in Schritt (b) gegeben ist durch die folgende Gleichung:

$$X_i = \Phi(r_i, P) = r_i P \text{ über } E_{a,b}(GF(q)),$$

die Signaturfunktion Sg_i in Schritt (d) gegeben ist durch die folgende Gleichung unter Verwendung von e_i, d_i, r_i und s_i :

$$y_i = Sg_i(e_i, d_i, s_i, r_i) = (d_i r_i + e_i s_i) \bmod k,$$

die Funktion V in Schritt (f) gegeben ist durch die folgende Gleichung:

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L) = (d_1 X_1 + \dots + d_L X_L + e_1 I_1 + \dots + e_L I_L) \text{ über } E_{a,b}(GF(q)), \text{ und}$$

die Funktion Γ in Schritt (g) gegeben ist durch die folgende Gleichung:

$$\text{wobei } Y = \sum_{i=1}^L y_i \bmod k.$$

15. Unterzeichnervorrichtung für ein System, bei dem jeder einer Reihe von Unterzeichnern $i = 1, \dots, L$, wobei L eine ganze Zahl gleich oder größer als zwei ist, eine digitale Signatur an einem in elektronischer Form vorliegenden Dokument m_i anbringt und ein Verifizierer die digitalen Signaturen en-bloc verifiziert, wobei Information enthaltend einen Parameter q für jeden der Unterzeichner i zur Erzeugung einer Signaturfunktion Sg_i und ein Parameter $\beta = G_1(q)$, der mit einer Funktion G_1 unter Verwendung des Parameters q erhalten wird, vorab veröffentlicht werden,

umfassend:

Speichermittel zur Speicherung der öffentlichen Parameter p und β des Systems, Identifikationsinformation ID_i des jeweiligen Unterzeichners i und einer ersten Zufallszahl s_i als seiner geheimen Information, G_2 -Funktionsmittel zur Erzeugung von Information $I_i = G_2(s_i, \beta)$ mit einer Funktion G_2 unter Verwendung des öffentlichen Parameters β und der ersten Zufallszahl s_i , wobei die Information I_i als öffentliche Information $\{ID_i, I_i, f_i, h_i\}$ des Unterzeichners zusammen mit einem Paar von Einweg-Funktionen f_i und h_i und Identifikationsinformation ID_i , die von dem jeweiligen Unterzeichner verwendet wird, veröffentlicht wird, 5

Zufallsgeneratormittel zur Erzeugung einer zweiten Zufallszahl r_i , Φ -Funktionsmittel zum Einsetzen des Parameters β und der zweiten Zufallszahl r_i in eine Funktion Φ zur Erzeugung von Information $X_i = \Phi(r_i, \beta)$, 10
zwei Einweg-Funktionsmittel zur Erzeugung von

$$e_i = f_i(X'_i, m'_i)$$

$$d_i = h_i(X'_i, m'_i) \quad 15$$

mit den beiden Einweg-Funktionen f_i und h_i unter Verwendung von Dokumentinformation m'_i enthaltend ein zu unterzeichnendes Dokument m_i und Information X'_i enthaltend die Information X_i ,
Signaturfunktionsmittel zur Erzeugung, für Information enthaltend e_i, d_i, s_i, r_i und y_{i-1} , einer Signatur 20

$$y_i = Sg_i(e_i, d_i, s_i, r_i, y_{i-1})$$

mit einer Signaturfunktion Sg_i , die unter Verwendung des Parameters q erzeugt wird, und Mittel zum Senden von Information $\{ID'_i, X'_i, m'_i, y_i\}$, als Information, welche Identifikationsinformation ID_i enthält, an den nächsten Unterzeichner $(i+1)$ in der Reihe von Unterzeichnern i von 1 bis L , wobei der letzte Unterzeichner L die Information $\{ID'_L, X'_L, m'_L, y_L\}$ an den Verifizierer als letzter Bestimmung sendet. 25

16. Unterzeichnervorrichtung für ein System, bei dem jeder von Unterzeichnern $i = 1, \dots, L$, wobei L eine ganze Zahl gleich oder größer als zwei ist, eine digitale Signatur an einem in elektronischer Form vorliegenden Dokument m'_i anbringt und ein Verifizierer die digitalen Signaturen en-bloc verifiziert, wobei Information enthaltend einen Parameter q für jeden der Unterzeichner i zur Erzeugung einer Signaturfunktion Sg_i und ein Parameter $\beta = G_1(q)$, der unter Verwendung des Parameters q mit einer Funktion G_1 erhalten wird, im voraus veröffentlicht werden, umfassend: 30

Speichermittel zur Speicherung der öffentlichen Parameter q und β des Systems, von Identifikationsinformation ID_i des jeweiligen Unterzeichners i und einer ersten Zufallszahl s_i als seiner geheimen Information, G_2 -Funktionsmittel zur Erzeugung von Information $I_i = G_2(s_i, \beta)$ mit einer Funktion G_2 unter Verwendung des öffentlichen Parameters β und der ersten Zufallszahl s_i , wobei die Information I_i zusammen mit einem Paar von Einweg-Funktionen f_i und h_i und Identifikationsinformation ID_i , die der jeweilige Unterzeichner i benutzt, als öffentliche Information $\{ID_i, I_i, f_i, h_i\}$ des Unterzeichners veröffentlicht wird, 35
Zufallsgeneratormittel zur Erzeugung einer zweiten Zufallszahl r_i , Φ -Funktionsmittel zum Einsetzen des Parameters β und der zweiten Zufallszahl r_i in eine Funktion Φ zur Erzeugung von Information $X_i = \Phi(r_i, \beta)$, 40
ein Paar von Einweg-Funktionsmitteln zur Erzeugung von

$$e_i = f_i(X'_i, m'_i)$$

$$d_i = h_i(X'_i, m'_i) \quad 45$$

mit dem Paar von Einweg-Funktionen f_i und h_i unter Verwendung der Dokumentinformation m'_i enthaltend ein zu unterzeichnendes Dokument m_i und Information X'_i enthaltend Information X_i , Signaturfunktionsmittel zur Erzeugung, für Information enthaltend e_i, d_i, s_i und r_i , einer Signatur 50

$$y_i = Sg_i(e_i, d_i, s_i, r_i)$$

mit der Signaturfunktion Sg_i , die unter Verwendung des Parameters q erzeugt wird, und Mittel zum Senden von Information $\{ID'_i, X'_i, m'_i, y_{i-1}\}$, als die Identifikationsinformation ID_i enthaltende Information an den Verifizierer. 55

17. Unterzeichnervorrichtung nach Anspruch 15, bei der

$$X'_i = (X'_{i-1}, X_i),$$

$$m'_i = (m'_{i-1}, m_i),$$

$$ID'_i = (ID'_{i-1}, ID_i),$$

$$X_0 = \text{leere Menge}$$

$$m_0 = \text{leere Menge}$$

$$ID'_0 = \text{leere Menge}$$

$$y_0 = 0, \text{ und}$$

bei Empfang von $\{ID'_{i-1}, X'_{i-1}, m'_{i-1}, y_{i-1}\}$ von dem Unterzeichner $(i-1)$ die Sendemittel Information $\{ID'_i, X'_i, m'_i, y_i\}$ an den nächsten Unterzeichner $(i+1)$ aussenden. 60

18. Unterzeichnervorrichtung nach Anspruch 17, bei der $m'_1 = m_1 = m, m_2 = m_3 = \dots = m = \text{leere Menge}$, die Unterzeichnervorrichtung des Unterzeichners i Information $\{ID'_{i-1}, X'_{i-1}, m'_{i-1}, y_{i-1}\}$ von dem Unterzeichner $(i-1)$ empfängt, und die Sendemittel Information $\{ID'_i, X'_i, m, y_i\}$ an den nächsten Unterzeichner $(i+1)$ aussenden. 65

19. Unterzeichnervorrichtung nach Anspruch 16, bei der $X'_i = X_i$, $m'_i = m_i$, und $ID'_i = ID_i$ und die Sendemittel Information $\{ID_i, X_i, m_i, y_i\}$, als die Information $\{ID'_i, X'_i, m'_i, y_i\}$, erzeugen und an den Verifizierer aussenden.

20. Unterzeichnervorrichtung nach Anspruch 17 oder 18, bei der $ID'_i = (ID'_{i-1}, ID_i)$ ersetzt wird durch $ID'_i = (ID'_{i-1}, I_i)$.

21. Unterzeichnervorrichtung nach Anspruch 17 oder 18, bei der, wenn p die Anzahl von Elementen einer Gruppe repräsentiert, ein Element g der Gruppe, bei dem eine Gruppenberechnung beginnt, durch den Parameter p repräsentiert wird und eine ganze Zahl bei der, wenn das Element g q -mal gruppenberechnet wird, die Berechnung zu g zurückkehrt, durch den Parameter q repräsentiert wird und diese Parameter $\{p, q, g\}$ als öffentliche Systeminformation veröffentlicht werden,

die G_2 -Funktionsmittel Mittel sind zur Berechnung der Information I_i durch Durchführen von s_i -maligen Gruppenberechnungen des Parameters g unter Verwendung des Parameters p , und
die Φ -Funktionsmittel Mittel sind zur Berechnung der Information X_i durch Durchführung von r_i -maligen Gruppenberechnungen des Parameters g unter Verwendung des Parameters p .

22. Unterzeichnervorrichtung nach Anspruch 21, bei der p und q Primzahlen sind, für die die Beziehung gilt $1 = p \bmod q$ und ein Grundelement von $(\mathbb{Z}/p\mathbb{Z})^*$ durch α repräsentiert wird, der Parameter $\beta = g$ durch folgende Gleichung mit der Funktion G_1 gegeben ist:

$$g = G_1(q) = \alpha^{(p-1)/q} \bmod p,$$

die G_2 -Funktionsmittel Mittel sind zur Berechnung der Funktion $G_2(s_i, g)$ durch die folgende Gleichung:

$$I_i = G_2(s_i, g) = g^{s_i} \bmod p$$

die Φ -Funktionsmittel Mittel sind zur Berechnung der Funktion Φ durch die folgende Gleichung:

$$X_i = \Phi(r_i, g) = g^{r_i} \bmod p, \text{ und}$$

die Signaturfunktionsmittel Mittel sind zur Berechnung der Signaturfunktion Sg_i durch die folgende Gleichung unter Verwendung von e_i, d_i, r_i, s_i, q und y_{i-1} :

$$y_i = Sg_i(e_i, d_i, s_i, r_i, y_{i-1}) = (y_{i-1} + d_i r_i + e_i s_i) \bmod q.$$

23. Unterzeichnervorrichtung nach Anspruch 19, bei der, wenn p die Anzahl von Elementen einer Gruppe repräsentiert, ein Element g der Gruppe, bei dem eine Gruppenberechnung beginnt, durch den Parameter β repräsentiert wird und eine ganze Zahl bei der, wenn das Element g q -mal gruppenberechnet wird, die Berechnung zu g zurückkehrt, durch den Parameter q repräsentiert wird, diese Parameter $\{p, q, g\}$ als öffentliche Systeminformation veröffentlicht werden,

die G_2 -Funktionsmittel Mittel sind zur Berechnung der Information I_i auf der Basis der Funktion $G_2(s_i, g)$, durch Durchführen von s_i -maligen Gruppenberechnungen des Parameters g unter Verwendung des Parameters p , und
die Φ -Funktionsmittel Mittel sind zur Berechnung der Information X_i auf der Basis der Funktion Φ durch Durchführen von r_i -maligen Gruppenberechnungen des Parameters g unter Verwendung des Parameters p .

24. Unterzeichnervorrichtung nach Anspruch 23, bei der p und q Primzahlen sind, für die die Beziehung gilt $1 = p \bmod q$ und ein Grundelement von $(\mathbb{Z}/p\mathbb{Z})^*$ durch α repräsentiert wird, die G_1 -Funktionsmittel zur Berechnung der Parameters $\beta = g$ Mittel sind zur Berechnung von

$$g = G_1(q) = \alpha^{(p-1)/q} \bmod p;$$

die G_2 -Funktionsmittel Mittel sind zur Berechnung der Funktion $G_2(s_i, g)$ durch die folgende Gleichung:

$$I_i = G_2(s_i, g) = g^{s_i} \bmod p$$

die Φ -Funktionsmittel Mittel sind zur Berechnung der Funktion Φ durch die folgende Gleichung:

$$X_i = \Phi(r_i, g) = g^{r_i} \bmod p, \text{ und}$$

die Signaturfunktionsmittel Mittel sind zur Berechnung der Signaturfunktion Sg_i durch die folgende Gleichung unter Verwendung von e_i, d_i, r_i, s_i und q :

$$y_i = Sg_i(e_i, d_i, s_i, r_i) = (d_i r_i + e_i s_i) \bmod q.$$

25. Unterzeichnervorrichtung nach Anspruch 17 oder 18, bei der der Parameter q ein Parameter eines Definitionsfeldes $GF(q)$ einer elliptischen Kurve $E_{a,b}(GF(q))$ ist, und, wenn ein Basispunkt einer Ordnung k auf der elliptischen Kurve durch den Parameter β repräsentiert wird und ein Parameter der elliptischen Kurve durch $a, b \in GF(q)$, diese Parameter $\{q, a, b, P, k\}$ als öffentliche Systeminformation veröffentlicht werden, und bei der

die G_2 -Funktionsmittel Mittel sind zur Berechnung der Information I_i auf der Basis der Funktion $G_2(s_i, P)$ durch Durchführen von s_i -maligen Gruppenberechnungen des Parameters p , und
die Φ -Funktionsmittel Mittel sind zur Berechnung der Information X_i auf der Basis der Funktion Φ durch Durchführen von r_i -maligen Gruppenberechnungen des Parameters P .

26. Unterzeichnervorrichtung nach Anspruch 25, bei der der Parameter $\beta = P$ auf der Basis der Funktion G_1 durch die folgende Gleichung gegeben ist:

$$G_1(q) = P \in E_{a,b}(GF(q));$$

5

die G_2 -Funktionsmittel Mittel sind zur Berechnung der Funktion $G_2(s_i, P)$ durch die folgende Gleichung:

$$I_i = G_2(s_i, P) = s_i P \text{ über } E_{a,b}(GF(q));$$

die Φ -Funktionsmittel Mittel sind zur Berechnung der Funktion Φ durch die folgende Gleichung:

10

$$X_i = \Phi(r_i, P) = r_i P \text{ über } E_{a,b}(GF(q)); \text{ und}$$

die Signaturfunktionsmittel Mittel sind zur Berechnung der Signaturfunktion Sg_i durch die folgende Gleichung unter Verwendung von e_i, d_i, r_i, s_i und y_{i-1} :

15

$$y_i = Sg_i(e_i, d_i, s_i, r_i, y_{i-1}) = (y_{i-1} + d_i r_i + e_i s_i) \bmod k.$$

27. Unterzeichnervorrichtung nach Anspruch 19, bei der der Parameter q ein Parameter eines Definitionsfeldes $GF(q)$ einer elliptischen Kurve $E_{a,b}(GF(q))$ ist, und, wenn ein Basispunkt einer Ordnung k auf der elliptischen Kurve durch den Parameter β repräsentiert wird und ein Parameter der elliptischen Kurve durch $a, b \in GF(q)$, diese Parameter $\{q, a, b, P, k\}$ als öffentliche Systeminformation veröffentlicht werden, und bei der:

20

die G_2 -Funktionsmittel Mittel sind zur Berechnung der Information I_i auf der Basis der Funktion $G_2(s_i, P)$ durch Durchführen von s_i -maligen Gruppenberechnungen des Parameters p , und

die Φ -Funktionsmittel Mittel sind zur Berechnung der Information X_i auf der Basis der Funktion Φ durch Durchführen von r_i -maligen Gruppenberechnungen des Parameters P .

25

28. Unterzeichnervorrichtung nach Anspruch 27, bei der:

der Parameter $\beta = P$ auf der Basis der Funktion G_1 gegeben ist durch die folgenden Gleichung:

$$G_1(q) = P \in E_{a,b}(GF(q));$$

30

die G_2 -Funktionsmittel Mittel sind zur Berechnung der Funktion $G_2(s_i, P)$ durch die folgende Gleichung:

$$I_i = G_2(s_i, P) = s_i P \text{ über } E_{a,b}(GF(q));$$

35

die Φ -Funktionsmittel Mittel sind zur Berechnung der Funktion Φ durch die folgende Gleichung;

$$X_i = \Phi(r_i, P) = r_i P \text{ über } E_{a,b}(GF(q)), \text{ und}$$

die Signaturfunktionsmittel Mittel sind zur Berechnung der Signaturfunktion Sg_i durch die folgende Gleichung unter Verwendung von e_i, d_i, r_i und s_i :

40

$$y_i = Sg_i(e_i, d_i, s_i, r_i) = (d_i r_i + e_i s_i) \bmod k.$$

29. Verifizierervorrichtung für ein System, bei dem jeder einer Reihe von Unterzeichnern $i = 1$ bis L , wobei L eine ganze Zahl gleich oder größer als zwei ist, eine digitale Signatur an einem in elektronischer Form vorliegenden Dokument m'_i anbringt und ein Verifizierer die digitale Signaturen der Unterzeichner en-bloc verifiziert, wobei Information enthaltend einen Parameter q für jeden der Unterzeichner i zur Erzeugung einer Signaturfunktion Sg_i und ein Parameter $\beta = G_1(q)$, der unter Verwendung des Parameters q mit einer Funktion G_1 erhalten wird, im voraus veröffentlicht werden, umfassend

45

ein Paar Einweg-Funktionsmittel zum Erhalt, aus öffentlicher Information $\{ID_i, I_i, f_i, h_i\}$, von Information I_i entsprechend Identifikationsinformation ID_i , die in ID'_L in Information $\{ID'_L, X'_L, m'_L, y_L\}$ enthalten ist, welche von dem letzten der Reihe von Unterzeichnern empfangen wird, sowie zum Erhalt von Einweg-Funktionen f_i und h_i und zur Berechnung von

50

$$e_i = f_i(X'_i, m'_i)$$

55

$$d_i = h_i(X'_i, m'_i)$$

unter Verwendung der Einweg-Funktionen f_i und h_i und Information X'_i und m'_i , die in den empfangenen Informationen X'_L und m'_L enthalten ist, V-Funktionsmittel zum Erhalt von X_i in der Information X'_i und zur Berechnung von

60

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L)$$

65

unter Verwendung der Funktion V , die eine Berechnung $(X_i * d_i)$ von d_i und X_i sowie eine Berechnung $(I_i * e_i)$ von e_i und I_i enthält,

Γ -Funktionsmittel zum Erhalt von $W = \Gamma(y_i * \beta)$ durch eine Funktion Γ , die eine Berechnung $(y_i * \beta)$ von y_i und β ent-

hält, und

Vergleichsmittel, die mit Z' und W beliefert werden für einen Vergleich, ob diese miteinander übereinstimmen, und, wenn sie miteinander übereinstimmen, zur Lieferung einer Ausgabe, die anzeigt, daß das empfangene Dokument (m_1, \dots, m_L) von den L Unterzeichnervorrichtungen ordnungsgemäß unterzeichnet wurde.

- 5 30. Verifizierervorrichtung für ein System, bei dem jeder von Unterzeichnern $i = 1$ bis L , wobei L eine ganze Zahl gleich oder größer als zwei ist, eine digitale Signatur an einem in elektronischer Form vorliegenden Dokument m'_i anbringt und ein Verifizierer die digitalen Signaturen der Unterzeichner en-bloc verifiziert, wobei Information enthaltend einen Parameter q für jeden der Unterzeichner i zur Erzeugung einer Signaturfunktion Sg_i und ein Parameter $\beta = G_1(q)$, der unter Verwendung des Parameters q mit einer Funktion G_1 erhalten wird, im voraus veröffentlicht werden, umfassend
- 10 ein Paar Einweg-Funktionsmittel zum Erhalt, aus öffentlicher Information $\{ID_i, I_i, f_i, h_i\}$, von Information I_i entsprechend Identifikationsinformation ID_i , die in ID_i in Information $\{ID'_i, X_i, m'_i, y_i\}$ enthalten ist, welche von den einzelnen Unterzeichnern i empfangen wird, sowie von Einweg-Funktionen f_i und h_i und zur Berechnung von $e_i = f_i(X'_i, m'_i)$
- 15 $d_i = h_i(X'_i, m'_i)$

unter Verwendung der Einweg-Funktionen f_i und h_i und der empfangenen Informationen X'_i und m'_i ;
V-Funktionsmittel zum Erhalt von X_i in der Information X'_i und zur Berechnung von

- 20 $Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L)$

unter Verwendung einer Funktion, die eine Berechnung $(X_i * d_i)$ von d_i und X_i sowie eine Berechnung $(I_i * e_i)$ von e_i und I_i enthält,

- 25 Γ -Funktionsmittel zum Erhalt von $W = \Gamma(Y * \beta)$ mittels einer Funktion Γ , die eine Berechnung $(Y * \beta)$ unter Verwendung von β und eines akkumulierten Werts Y von y_1 bis y_L enthält, und
Vergleichsmittel, die mit Z' und W beliefert werden, um zu prüfen, ob diese beiden übereinstimmen, und, wenn sie miteinander übereinstimmen, zur Lieferung einer Ausgabe, die anzeigt, daß das empfangene Dokument (m_1, \dots, m_L) von L Unterzeichnern ordnungsgemäß unterzeichnet wurde.

- 30 31. Verifizierervorrichtung nach Anspruch 29, bei der

- $X'_i = (X'_{i-1}, X_i)$,
 $m'_i = (m'_{i-1}, m_i)$,
 $ID'_i = (ID'_{i-1}, ID_i)$,
 $X_0 =$ leere Menge
35 $m'_0 =$ leere Menge
 $ID'_0 =$ leere Menge
 $y_0 =$ leere Menge.

32. Verifizierervorrichtung nach Anspruch 31, bei der $m'_1 = m_1 = m$; $m_2 = m_3 = \dots = m_L =$ leere Menge und die Verifizierervorrichtung unterzeichnete Information $\{ID_L, X_L, m_L, y_L\}$ direkt von dem letzten Unterzeichner L empfängt.

- 40 33. Verifizierervorrichtung nach Anspruch 30, bei der $X'_i = X_i$, $m'_i = m_i$, und $ID'_i = ID_i$.

34. Verifizierervorrichtung nach Anspruch 31 oder 32, bei der $ID'_i = (ID'_{i-1}, ID_i)$ ersetzt ist durch $ID'_i = (ID'_{i-1}, I_i)$.

35. Verifizierervorrichtung nach Anspruch 31 oder 32, bei der, wenn p die Anzahl von Elementen einer Gruppe repräsentiert, ein Element g der Gruppe, bei dem eine Gruppenberechnung beginnt, durch den Parameter β repräsentiert wird und eine ganze Zahl, bei der, wenn das Element g q -mal gruppenberechnet wird, die Berechnung zu g zurückkehrt, durch den Parameter q repräsentiert wird und diese Parameter $\{p, q, g\}$ als öffentliche Systeminformation veröffentlicht werden,

- 45 die V-Funktionsmittel Mittel sind zum Erhalt von Z' durch sequentielle Berechnung von Werten $(X_i * d_i)$, erhalten durch d_i -malige Multikomponenten-Gruppenberechnungen von X_i , und Werten $(I_i * e_i)$, erhalten durch e_i -malige Multikomponenten-Gruppenberechnungen von I_i für jedes i von 1 bis L , und

- 50 die Γ -Funktionsmittel Mittel sind zum Erhalt von W durch Durchführen von y_L -maligen Gruppenberechnungen des Parameters g unter Verwendung der empfangenen Information y_L und der öffentlichen Informationen p und q .

36. Verifizierervorrichtung nach Anspruch 35, bei der p und q Primzahlen sind, für die die Beziehung gilt $1 = p \bmod q$, und, wenn α ein Grundelement von $(\mathbb{Z}/p\mathbb{Z})^*$ repräsentiert,
55 der Parameter $\beta = g$ durch folgende Gleichung mit der Funktion G_1 gegeben ist:

$$g = G_1(q) = \alpha^{(p-1)q} \bmod p;$$

die V-Funktionsmittel Mittel sind zur Berechnung der Funktion V durch die folgende Gleichung:

- 60 $Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L) = X_1^d \mid I_1^e \mid \dots \mid X_L^d \mid I_L^e \bmod p$, und

die Γ -Funktionsmittel Mittel sind zur Berechnung der Funktion Γ durch die folgende Gleichung:

- 65 $W = \Gamma(y_i * g) = g^{y_L} \bmod p$.

37. Verifizierervorrichtung nach Anspruch 33, bei der, wenn p die Anzahl von Elementen einer Gruppe repräsentiert, ein Element g der Gruppe, bei dem eine Gruppenberechnung beginnt, durch den Parameter β repräsentiert wird

und eine ganze Zahl, bei der, wenn das Element g q -mal gruppenberechnet wird, die Berechnung zu g zurückkehrt, durch den Parameter q repräsentiert wird und diese Parameter $\{p, q, g\}$ als öffentliche Systeminformation veröffentlicht werden,

die V-Funktionsmittel Mittel sind zum Erhalt von Z' durch sequentielle Berechnung von Werten $(X_i * d_i)$, erhalten durch d_i -malige Multikomponenten-Gruppenberechnungen von X_i , und Werten $(I_i * e_i)$, erhalten durch e_i -malige Multikomponenten-Gruppenberechnungen von I_i , für jedes i von 1 bis L , und

die Γ -Funktionsmittel Mittel sind zum Berechnen eines akkumulierten Werts Y unter Verwendung von L empfangenen Informationen y_i und zur Erzeugung, als W , eines Wertes $(g * Y)$, erhalten durch Y -maliges Berechnen des Parameters g unter Verwendung von Y und der öffentlichen Informationen p und q .

38. Verifizierervorrichtung nach Anspruch 37, bei der p und q Primzahlen sind, für die die Beziehung gilt $1 = p \bmod q$, und, wenn α ein Grundelement von $(\mathbb{Z}/p\mathbb{Z})^*$ repräsentiert, der Parameter $\beta = g$ durch folgende Gleichung mit der Funktion G_1 gegeben ist:

$$g = G_1(q) = \alpha^{(p-1)/q} \bmod p;$$

die V-Funktionsmittel Mittel sind zur Berechnung der Funktion V durch die folgende Gleichung:

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L) = X_1^{d_1} I_1^{e_1} \dots X_L^{d_L} I_L^{e_L} \bmod p, \text{ und}$$

die Γ -Funktionsmittel Mittel sind zur Berechnung des akkumulierten Werts Y durch folgende Gleichung:

$$Y = \sum_{i=1}^L y_i \bmod q$$

und zur Berechnung der Funktion Γ durch die folgende Gleichung:

$$W = \Gamma(Y * g) = g^Y \bmod p.$$

39. Verifizierervorrichtung nach Anspruch 31 oder 32, bei der der Parameter q ein Parameter eines Definitionsfelds $\text{GF}(q)$ einer elliptischen Kurve $E_{a,b}(\text{GF}(q))$ ist, und, wenn ein Basispunkt einer Ordnung k der elliptischen Kurve durch den Parameter β repräsentiert wird und ein Parameter der elliptischen Kurve durch $a, b \in \text{GF}(q)$, diese Parameter $\{q, a, b, P, k\}$ als öffentliche Systeminformation veröffentlicht werden, und bei der

die V-Funktionsmittel Mittel sind zum Erhalt von Z' durch sequentielle Berechnung von Werten $(X_i * d_i)$, erhalten durch d_i -malige Multikomponenten-Gruppenberechnung von X_i , und Werten $(I_i * e_i)$, erhalten durch e_i -malige Multikomponenten-Gruppenberechnung von I_i , für jedes i von 1 bis L , und

die Γ -Funktionsmittel Mittel sind zur Berechnung von W durch y_L -malige Berechnungen des Parameters P auf der elliptischen Kurve unter Verwendung der empfangenen Information y_L und der öffentlichen Information P .

40. Verifizierervorrichtung nach Anspruch 39, bei der:

die Funktion G_1 zur Berechnung des Parameters $\beta = P$ gegeben ist durch folgende Gleichung:

$$G_1(q) = P \in E_{a,b}(\text{GF}(q));$$

die V-Funktionsmittel Mittel sind zur Berechnung der Funktion V durch die folgende Gleichung:

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L) = (d_1 X_1 + \dots + d_L X_L + e_1 I_1 + \dots + e_L I_L) \text{ über } E_{a,b}(\text{GF}(q)), \text{ und}$$

die Γ -Funktionsmittel Mittel sind zur Berechnung der Funktion Γ durch die folgende Gleichung:

$$W = \Gamma(y_L * P) = y_L P \text{ über } E_{a,b}(\text{GF}(q)).$$

41. Verifizierervorrichtung nach Anspruch 33, bei der der Parameter q ein Parameter eines Definitionsfelds $\text{GF}(q)$ einer elliptischen Kurve $E_{a,b}(\text{GF}(q))$ ist, und, wenn ein Basispunkt einer Ordnung k auf der elliptischen Kurve durch den Parameter β repräsentiert wird und ein Parameter der elliptischen Kurve durch $a, b \in \text{GF}(q)$, diese Parameter $\{q, a, b, P, k\}$ als öffentliche Systeminformation veröffentlicht werden, und

die V-Funktionsmittel Mittel sind zum Erhalt von Z' durch sequentielle Berechnung von Werten $(X_i * d_i)$, erhalten durch d_i -malige Multikomponenten-Gruppenberechnungen von X_i , und Werten $(I_i * e_i)$, erhalten durch e_i -malige Multikomponenten-Gruppenberechnungen von I_i , für jedes i von 1 bis L , und

die Γ -Funktionsmittel Mittel sind zur Berechnung von W durch y_L -malige Berechnungen des Parameters P auf der elliptischen Kurve unter Verwendung der empfangenen Information y_L und der öffentlichen Information P .

42. Verifizierervorrichtung nach Anspruch 41, bei der die Funktion G_1 zur Berechnung des Parameters $\beta = P$ gegeben ist durch folgende Gleichung:

$$G_1(q) = P \in E_{a,b}(\text{GF}(q));$$

die V-Funktionsmittel Mittel sind zur Berechnung der Funktion V durch die folgende Gleichung:

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L) = (d_1 X_1 + \dots + d_L X_L + e_1 I_1 + \dots + e_L I_L) \text{ über } E_{a,b}(\text{GF}(q)), \text{ und}$$

die Γ -Funktionsmittel Mittel sind zur Berechnung der Funktion Γ durch die folgende Gleichung:

$$W = \Gamma(Y * P) = YP \text{ über } E_{a,b}(GF(q)),$$

wobei

$$Y = \sum_{i=1}^L y_i \text{ mod } k.$$

43. Speichermedium für eine Unterzeichnervorrichtung, auf dem ein Programm gespeichert ist, welches in der Unterzeichnervorrichtung in einem System verwendet wird, bei dem:

jeder einer Reihe von Unterzeichnern i eine digitale Signatur an einem in elektronischer Form vorliegenden Dokument m'_i anbringt und ein Verifizierer die digitalen Signaturen en-bloc verifiziert, wobei $i = 1, \dots, L$ und L eine ganze Zahl gleich oder größer als zwei ist, Information enthaltend einen Parameter q für jeden Unterzeichner i zur Erzeugung einer Signaturfunktion S_{g_i} und einen Parameter $\beta = G_1(q)$, der unter Verwendung des Parameters q mit einer Funktion G_1 erhalten wird, vorab veröffentlicht wird, und Information $I_i = G_2(s_i, \beta)$, vorab erzeugt von jedem Unterzeichner i unter Verwendung des öffentlichen Parameters β und einer geheimen Zufallszahl s_i , ein Paar von Einweg-Funktionen f_i und h_i zur Verwendung durch den jeweiligen Unterzeichner i und Identifikationsinformation ID_i vorab als öffentliche Unterzeichnerinformation $\{ID_i, f_i, h_i\}$ veröffentlicht werden, wobei das Programm die Schritte enthält:

- (a) Erzeugen einer zweiten Zufallszahl r_i ,
- (b) Einsetzen des Parameters β und der zweiten Zufallszahl r_i in eine Funktion Φ zum Erzeugen von Information $X_i = \Phi(r_i, \beta)$,
- (c) Erzeugen von

$$e_i = f_i(X'_i, m'_i)$$

$$d_i = h_i(X'_i, m'_i)$$

mit den Einweg-Funktionen f_i und h_i unter Verwendung von Dokumentinformation m'_i enthaltend das zu unterzeichnende Dokument m_i und der Information X'_i enthaltend die Information X_i ,

- (d) Erzeugen, für Information enthaltend e_i, d_i, s_i, r_i und y_{i-1} , einer Signatur

$$y_i = S_{g_i}(e_i, d_i, s_i, r_i, y_{i-1})$$

mit einer Signaturfunktion S_{g_i} , die unter Verwendung des Parameters q erzeugt wird, und

- (e) Senden von Information $\{ID'_i, X'_i, m'_i, y_i\}$, enthaltend die Identifikationsinformation ID'_i als Identifikationsinformation ID'_i , an den nächsten Unterzeichner $(i+1)$, wobei der letzte Unterzeichner L die Information $\{ID'_L, X'_L, m'_L, y_L\}$ an den Verifizierer als letzte Bestimmung sendet.

44. Speichermedium für eine Unterzeichnervorrichtung, auf dem ein Programm gespeichert ist, das in der Unterzeichnervorrichtung in einem System verwendet wird, bei dem: jeder von Unterzeichnern i eine digitale Signatur an einem in elektronischer Form vorliegenden Dokument m'_i anbringt und ein Verifizierer die digitalen Signaturen en-bloc verifiziert, wobei $i = 1, \dots, L$ und L eine ganze Zahl gleich oder größer als zwei ist, wobei Information enthaltend einen Parameter q für jeden Unterzeichner i zur Erzeugung einer Signaturfunktion S_{g_i} und einen Parameter $\beta = G_1(q)$, der mit einer Funktion G_1 unter Verwendung des Parameters q erhalten wird, vorab veröffentlicht werden, und Information $I_i = G_2(s_i, \beta)$, vorab erzeugt durch den jeweiligen Unterzeichner i unter Verwendung des öffentlichen Parameters β und einer geheimen Zufallszahl s_i , ein Paar von Einweg-Funktionen f_i und h_i zur Verwendung durch den jeweiligen Unterzeichner i und Identifikationsinformation ID_i als öffentliche Unterzeichnerinformation $\{ID_i, I_i, f_i, h_i\}$ vorab veröffentlicht werden, wobei das Programm die Schritte umfaßt:

- (a) Erzeugen einer zweiten Zufallszahl r_i ,
- (b) Einsetzen des Parameters β und der zweiten Zufallszahl r_i in eine Funktion Φ zum Erzeugen von Information $X_i = \Phi(r_i, \beta)$,
- (c) Erzeugen von

$$e_i = f_i(X'_i, m'_i)$$

$$d_i = h_i(X'_i, m'_i)$$

mit den Einweg-Funktionen f_i und h_i unter Verwendung von Dokumentinformation m'_i enthaltend das zu unterzeichnende Dokument m_i und der Information X'_i ,

- (d) Erzeugen, für Information enthaltend e_i, d_i, s_i und r_i , einer Signatur

$$y_i = S_{g_i}(e_i, d_i, s_i, r_i)$$

mit einer Signaturfunktion S_{g_i} , die unter Verwendung des Parameters q erzeugt wird, und

- (e) Senden von Information $\{ID'_i, X'_i, m'_i, y_i\}$, enthaltend die Identifikationsinformation ID'_i als Identifikationsinformation ID'_i an den Verifizierer.

45. Verfahren nach Anspruch 43, bei dem in dem Programm:

$X'_i = (X'_{i-1}, X_i)$,

$m'_i = (m'_{i-1}, m_i)$

$ID'_i = (ID'_{i-1}, ID_i)$,

X_0 = leere Menge

m_0 = leere Menge

ID_0 = leere Menge

$y_0 = 0$, und

der Unterzeichner i von dem Unterzeichner $(i-1)$ unterzeichnete Information $\{ID'_{i-1}, X'_{i-1}, m'_{i-1}, y_{i-1}\}$ empfängt und dann die Schritte (a) bis (d) auf der Basis der empfangenen Information ausführt.

46. Speichermedium nach Anspruch 45, bei dem in dem Programm: $m'_1 = m_1 = m$; $m_2 = m_3 = \dots = m_L$ = leere Menge, und der Unterzeichner i von dem Unterzeichner $(i-1)$ Information $\{ID'_{i-1}, X'_{i-1}, m, y_{i-1}\}$ empfängt und dann auf der Basis der empfangenen Information im Schritt (e) Information $\{ID'_i, X'_i, m, y_i\}$ an den nächsten Unterzeichner $(i+1)$ aussendet.

47. Speichermedium nach Anspruch 44, bei dem in dem Programm: $X'_i = X_i$, $m'_i = m_i$, und $ID'_i = ID_i$; und der Unterzeichner i die Schritte (a) bis (e) ausführt und dadurch Information $\{ID_i, X_i, y_i\}$ als die Information $\{ID'_i, X'_i, m'_i, y_i\}$ erzeugt und einzeln an den Verifizierer sendet.

48. Speichermedium nach Anspruch 45 oder 46, bei dem in dem Programm $ID'_i = (ID'_{i-1}, ID_i)$ ersetzt wird durch $ID'_i = (ID'_{i-1}, I_i)$.

49. Speichermedium nach Anspruch 45 oder 46, bei dem in dem Programm: wenn die Anzahl von Elementen einer Gruppe mit p bezeichnet wird, ein Element g der Gruppe, bei dem eine Gruppenberechnung beginnt, durch den Parameter β repräsentiert ist und eine ganze Zahl, bei der, wenn das Element g q -mal gruppenberechnet wird, die Rechnung zu g zurückkehrt, durch den Parameter q repräsentiert wird, diese Parameter $\{p, q, g\}$ als öffentliche Systeminformation veröffentlicht werden,

die Information I_i vorab durch s_i -malige Gruppenberechnung des Parameters g unter Verwendung des Parameters p durchgeführt wird, und

der Schritt (b) ein Schritt ist zum Erhalt von X_i durch r_i -malige Gruppenberechnungen des Parameters g unter Verwendung des Parameters p .

50. Speichermedium nach Anspruch 49, bei dem in dem Programm: p und q Primzahlen sind, zwischen denen die Beziehung $1 = p \bmod q$ gilt, und, wenn ein Grundelement von $(\mathbb{Z}/p\mathbb{Z})^*$ durch α repräsentiert wird, der Parameter $\beta = g$ vorab mit der Funktion G_1 durch folgende Gleichung gegeben ist:

$$g = G_1(q) = \alpha^{(p-1)/q} \bmod p;$$

die Information I_i vorab mit der Funktion G_2 durch folgende Gleichung gegeben ist:

$$I_i = G_2(s_i, g) = s_i \bmod p;$$

Schritt (d) ein Schritt der Berechnung der Information X_i mit der Funktion Φ durch folgende Gleichung ist:

$$X_i = \Phi(r_i, g) = g^{r_i} \bmod p, \text{ und}$$

Schritt (d) ein Schritt der Berechnung der Signaturfunktion Sg_i durch folgende Gleichung unter Verwendung von e_i , d_i , r_i , s_i , q und y_{i-1} ist:

$$y_i = Sg_i(e_i, d_i, s_i, r_i, y_{i-1}) = (y_{i-1} + d_i r_i + e_i s_i) \bmod q.$$

51. Speichermedium nach Anspruch 47, bei dem in dem Programm: wenn man die Anzahl von Elementen einer Gruppe mit p bezeichnet, ein Element g der Gruppe, bei dem eine Gruppenberechnung beginnt, durch den Parameter β repräsentiert wird und eine ganze Zahl, bei der, wenn das Element g q -mal gruppenberechnet wird, die Rechnung zu g zurückkehrt, durch den Parameter q repräsentiert wird, diese Parameter $\{p, q, g\}$ als öffentliche Systeminformation veröffentlicht werden,

die Information I_i vorab mit der Funktion $G_2(s_i, g)$ durch s_i -malige Gruppenberechnungen von g unter Verwendung von p erhalten wird, und

Schritt (b) ein Schritt zum Erhalt der Information X mit der Funktion $\Phi(r_i, g)$ durch r_i -malige Gruppenberechnungen von g unter Verwendung von p ist.

52. Speichermedium nach Anspruch 51, bei dem in dem Programm: p und q Primzahlen sind, zwischen denen die Beziehung gilt $1 = p \bmod q$, und, wenn ein Grundelement von $(\mathbb{Z}/p\mathbb{Z})^*$ durch α repräsentiert wird, der Parameter $\beta = g$ vorab mit der Funktion G_1 durch folgende Gleichung gegeben ist:

$$g = G_1(q) = \alpha^{(p-1)/q} \bmod p,$$

die Information I_i vorab mit der Funktion G_2 durch folgende Gleichung gegeben ist:

$$I_i = G_2(s_i, g) = g^{s_i} \bmod p,$$

Schritt (b) ein Schritt der Berechnung der Information X_i mit der Funktion Φ durch folgende Gleichung ist:

$X_i = \Phi(r_i, g) = g^{r_i} \bmod p$, und

Schritt (d) ein Schritt der Berechnung der Signaturfunktion Sg_i durch folgende Gleichung unter Verwendung von e_i , d_i , r_i , s_i und q ist:

$$y_i = Sg_i(e_i, d_i, s_i, r_i) = (d_i r_i + e_i s_i) \bmod q.$$

53. Speichermedium nach Anspruch 45 oder 46, bei dem in dem Programm: der Parameter q ein Parameter eines Definitionsfeldes $GF(q)$ einer elliptischen Kurve $E_{a,b}(GF(q))$ ist, und, wenn ein Basispunkt einer Ordnung k auf der elliptischen Kurve durch den Parameter β repräsentiert ist und ein Parameter der elliptischen Kurve durch $a, b \in GF(q)$, diese Parameter $\{q, a, b, P, k\}$ als öffentliche Systeminformation veröffentlicht werden, und bei dem die Information I_i vorab mit der Funktion $G_2(s_i, P)$ durch Durchführen von s_i -maligen Gruppenberechnungen des Parameters P erhalten wird, und

Schritt (b) ein Schritt zum Erhalt der Information X_i mit der Funktion Φ durch Durchführen von r_i -maligen Gruppenberechnungen des Parameters P ist.

54. Speichermedium nach Anspruch 53, bei dem in dem Programm: der Parameter $\beta = P$ auf der Basis der Funktion G_1 durch folgende Gleichung gegeben ist:

$$G_1(q) = P \in E_{a,b}(GF(q)),$$

die Information I_i mit der Funktion G_2 durch folgende Gleichung errechnet wird:

$$I_i = G_2(s_i, P) = s_i P \text{ über } E_{a,b}(GF(q)),$$

Schritt (b) ein Schritt der Berechnung der Information X_i mit der Funktion Φ durch folgende Gleichung ist:

$$X_i = \Phi(r_i, P) = r_i P \text{ über } E_{a,b}(GF(q)), \text{ und}$$

Schritt (d) ein Schritt der Berechnung der Signaturfunktion Sg_i durch die folgende Gleichung unter Verwendung von e_i , d_i , r_i , s_i und y_{i-1} ist:

$$y_i = Sg_i(e_i, d_i, s_i, r_i, y_{i-1}) = (y_{i-1} + d_i r_i + e_i s_i) \bmod k.$$

55. Speichermedium nach Anspruch 47, bei dem in dem Programm: der Parameter q ein Parameter eines Definitionsfeldes $GF(q)$ einer elliptischen Kurve $E_{a,b}(GF(q))$ ist, und, wenn ein Basispunkt einer Ordnung k auf der elliptischen Kurve durch den Parameter β repräsentiert ist und ein Parameter der elliptischen Kurve durch $a, b \in GF(q)$, diese Parameter $\{g, a, b, P, k\}$ als öffentliche Systeminformation veröffentlicht werden, und bei dem die Information I_i vorab mit der Funktion $G_2(s_i, P)$ erhalten wird durch Durchführen von s_i -maligen Gruppenberechnungen des Parameters P , und

Schritt (b) ein Schritt zum Erhalt der Information X_i mit der Funktion Φ durch Durchführen von r_i -maligen Gruppenberechnungen des Parameters P ist.

56. Speichermedium nach Anspruch 55, bei dem in dem Programm: der Parameter $\beta = P$ gegeben ist auf der Basis der Funktion G_1 durch die folgende Gleichung:

$$G_1(q) = P \in E_{a,b}(GF(q)),$$

die Information I_i mit der Funktion G_2 durch die folgende Gleichung berechnet wird:

$$I_i = G_2(s_i, P) = s_i P \text{ über } E_{a,b}(GF(q)),$$

Schritt (b) ein Schritt der Berechnung der Information X_i mit der Funktion Φ durch folgende Gleichung ist:

$$X_i = \Phi(r_i, P) = r_i P \text{ über } E_{a,b}(GF(q)), \text{ und}$$

Schritt (d) ein Schritt der Berechnung der Signaturfunktion Sg_i unter Verwendung von e_i , d_i , r_i und s_i unter Verwendung der folgenden Gleichung ist:

$$y_i = Sg_i(e_i, d_i, s_i, r_i) = (d_i r_i + e_i s_i) \bmod k.$$

57. Speichermedium für eine Verifizierungsvorrichtung, auf dem ein Programm gespeichert ist, das in der Verifizierungsvorrichtung in einem System verwendet wird, bei dem: jeder einer Reihe von Unterzeichnern i eine digitale Signatur an einem in elektronischer Form vorliegenden Dokument m_i anbringt und ein Verifizierer die digitalen Signaturen en-bloc verifiziert, wobei $i = 1, \dots, L$ und L eine ganze Zahl gleich oder größer als zwei ist, und Information enthaltend einen Parameter q für jeden Unterzeichner i zur Erzeugung einer Signaturfunktion Sg_i und einen Parameter $\beta = G_1(q)$, der mit einer Funktion G_1 unter Verwendung des Parameters q erhalten wird, vorab veröffentlicht wird,

wobei das Programm die Schritte umfaßt:

(a) Gewinnen, aus öffentlicher Information $\{ID_i, I_i, f_i, h_i\}$, von Information I_i entsprechend Identifikationsin-

formation ID_i , die in Information ID'_i in Information $\{ID'_L, X'_L, m'_L, y_L\}$ enthalten ist, welche von dem letzten Unterzeichner L der Reihe von Unterzeichnern empfangen wird, sowie von Einweg-Funktionen f_i und h_i , und Berechnen von

$$e_i = f_i(X'_i, m'_i) \quad 5$$

$$d_i = h_i(X'_i, m'_i)$$

unter Verwendung der Einweg-Funktionen f_i und h_i sowie von Information X'_i und m'_i , die in den empfangenen Informationen X'_L und m'_L enthalten sind, 10

(b) Gewinnen von X_i aus der Information X'_i und Berechnen von

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L)$$

durch eine Funktion V , enthaltend eine Berechnung von $(X_i * d_i)$ aus d_i und X_i sowie eine Berechnung von $(I_i * e_i)$ aus e_i und I_i ; 15

(c) Gewinnen von $W = \Gamma((y_i * \beta))$ durch eine Funktion Γ , enthaltend eine Berechnung von $(y_i * \beta)$ aus y_i und β ; und

(d) Empfangen der Werte Z' und W und anschließendes Prüfen, ob sie übereinstimmen, und, wenn sie übereinstimmen, Liefern einer Ausgabe, die anzeigt, daß die empfangenen Dokumente (m_1, \dots, m_L) von den L Unterzeichnern ordnungsgemäß unterzeichnet wurden. 20

58. Speichermedium für eine Verifizierervorrichtung, auf dem ein Programm gespeichert ist, welches in der Verifizierervorrichtung in einem System verwendet wird, bei dem: jeder von Unterzeichnern i eine digitale Signatur an einem in elektronischer Form vorliegenden Dokument m'_i anbringt und ein Verifizierer die digitalen Signaturen en bloc verifiziert, wobei $i = 1, \dots, L$ und L eine ganze Zahl gleich oder größer als zwei ist, und wobei Information enthaltend einen Parameter q für jeden Unterzeichner i zur Erzeugung einer Signaturfunktion Sg_i und ein Parameter $\beta = G_1(q)$, der unter Verwendung des Parameters q mit einer Funktion G_1 erhalten wird, vorab veröffentlicht werden, wobei das Programm die Schritte umfaßt: 25

(a) Gewinnen, aus öffentlicher Information $\{ID_i, I_i, f_i, h_i\}$, von Information I_i entsprechend Identifikationsinformation ID_i , die in ID'_i in der Information $\{ID'_i, X'_i, m'_i, y_i\}$ enthalten ist, die von jedem der Unterzeichner i empfangen wird, sowie von Einweg-Funktionen f_i und h_i und Berechnen von 30

$$e_i = f_i(X'_i, m'_i)$$

$$d_i = h_i(X'_i, m'_i) \quad 35$$

unter Verwendung der Einweg-Funktionen f_i und h_i und den empfangenen Informationen X'_i und m'_i ;

(b) Gewinnen von X_i in der Information X'_i und Berechnen von

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L) \quad 40$$

durch eine Funktion V , enthaltend eine Berechnung von $(X_i * d_i)$ aus d_i und X_i und eine Berechnung von $(I_i * e_i)$ aus e_i und I_i ;

(c) Gewinnen von $W = \Gamma(Y * \beta)$ durch eine Funktion Γ , enthaltend eine Berechnung von $(Y * \beta)$ unter Verwendung von β und eines akkumulierten Werts Y aus y_i bis y_L , und 45

(d) Empfangen der Werte Z' und W und anschließendes Prüfen, ob sie übereinstimmen, und, falls sie übereinstimmen, Liefern einer Ausgabe, die anzeigt, daß die empfangenen Dokumente (m_1, \dots, m_L) von L Unterzeichnern ordnungsgemäß unterzeichnet wurden.

59. Speichermedium nach Anspruch 57, bei dem in dem Programm:

$$X'_i = (X'_{i-1}, X_i), \quad 50$$

$$m'_i = (m'_{i-1}, m_i),$$

$$ID'_i = (ID'_{i-1}, ID_i),$$

$$X'_0 = \text{leere Menge}$$

$$m'_0 = \text{leere Menge}$$

$$ID'_0 = \text{leere Menge} \quad 55$$

$$y_0 = 0, \text{ und,}$$

wenn der Verifizierer unterzeichnete Information $\{ID'_L, X'_L, m'_L, y_L\}$ von dem letzten Unterzeichner L empfängt, der Verifizierer die Schritte (a) bis (d) auf der Basis der empfangenen Information ausführt.

60. Speichermedium nach Anspruch 59, bei dem in dem Programm: $m'_1 = m_1 = m$; $m'_2 = m_2 = m_3 = \dots = m_L = \text{leere Menge}$, und der Verifizierer unterzeichnete Information $\{ID'_L, X'_L, m, y_L\}$ von dem Unterzeichner L empfängt und die Schritte (a) bis (d) auf der Basis der empfangenen Information ausführt. 60

61. Speichermedium nach Anspruch 58, bei dem in dem Programm: $X'_i = X_i$, $m'_i = m_i$, und $ID'_i = ID_i$; und der Verifizierer unterzeichnete Information $\{ID_i, X_i, m_i, y_i\}$ gesondert von jedem einzelnen Unterzeichner i empfängt und die Schritte (a) bis (d) ausführt.

62. Speichermedium nach Anspruch 59 oder 60, bei dem in dem Programm $ID'_i = (ID'_{i-1}, ID_i)$ ersetzt ist durch $ID'_i = (ID'_{i-1}, I_i)$. 65

63. Speichermedium nach Anspruch 59 oder 60, bei dem in dem Programm, wenn die Anzahl von Elementen einer Gruppe mit p bezeichnet wird, ein Element g der Gruppe, bei dem eine Gruppenberechnung beginnt, durch den Pa-

parameter β repräsentiert wird und, eine ganze Zahl bei der, wenn das Element g q -mal gruppenberechnet wird, die Rechnung zu g zurückkehrt, durch den Parameter q repräsentiert wird und diese Parameter $\{p, q, g\}$ als öffentliche Systeminformation veröffentlicht werden,

der Schritt (b) ein Schritt zum Erhalt von Z' durch sequentielle Berechnung von Werten $(X_i * d_i)$, erhalten durch d_i -malige Multikomponenten-Gruppenberechnungen von X_i und Werten $(I_i * e_i)$, erhalten durch e_i -malige Multikomponenten-Gruppenberechnungen von I_i , für jedes i von 1 bis L ist, und
 5 der Schritt (c) ein Schritt ist zum Erhalt von W durch Durchführen von y_L -maligen Gruppenberechnungen des Parameters g unter Verwendung des empfangenen y_L und der öffentlichen Informationen p und q .

64. Speichermedium nach Anspruch 63, bei dem in dem Programm: p und q Primzahlen sind, für die die Beziehung
 10 gilt $1 = p \bmod q$, und, wenn ein Grundelement von $(\mathbb{Z}/p\mathbb{Z})^\times$ durch α repräsentiert wird, der Parameter $\beta = g$ im voraus gegeben ist durch die folgende Gleichung mit der Funktion G_1 :

$$g = G_1(q) = \alpha^{(p-1)q} \bmod p,$$

15 Schritt (b) ein Schritt ist zur Berechnung der Funktion V durch die Gleichung:

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L) = X_1^d I_1^e \dots X_L^d I_L^e \bmod p, \text{ und}$$

Schritt (c) ein Schritt ist der Berechnung der Funktion Γ durch die folgende Gleichung:

$$20 \quad W = \Gamma(y_i * g) = g^{y_L} \bmod p.$$

65. Speichermedium nach Anspruch 61, bei dem in dem Programm, wenn die Anzahl von Elementen einer Gruppe mit p bezeichnet wird, ein Element g der Gruppe, bei dem eine Gruppenberechnung beginnt, durch den Parameter β repräsentiert wird und eine ganze Zahl bei der, wenn das Element g q -mal gruppenberechnet wird, die Rechnung zu g zurückkehrt, durch den Parameter q repräsentiert wird, diese Parameter $\{p, q, g\}$ als öffentliche Systeminformation veröffentlicht werden,

Schritt (b) ein Schritt ist zum Erhalt von Z' durch sequentielles Berechnen von Werten $(X_i * d_i)$, erhalten durch d_i -malige Multikomponenten-Gruppenberechnungen von X_i , und Werten $(I_i * e_i)$, erhalten durch e_i -malige Multikomponenten-Gruppenberechnungen von I_i , für jedes i von 1 bis L , und
 30 Schritt (c) ein Schritt ist der Berechnung eines akkumulierten Werts Y unter Verwendung von L empfangenen Informationen y_i und der öffentlichen Information q , sowie der Berechnung, als W , eines Werts $(g * Y)$, der erhalten wird durch Y -maliges Operieren oder Berechnen des Parameters g unter Verwendung der öffentlichen Informationen p und q .

66. Speichermedium nach Anspruch 65, bei dem in dem Programm: p und q Primzahlen sind, für die die Beziehung
 35 gilt $1 = p \bmod q$, und, wenn ein Grundelement von $(\mathbb{Z}/p\mathbb{Z})^\times$ durch α repräsentiert wird, der Parameter $\beta = g$ gegeben ist durch die folgende Gleichung mit der Funktion G_1 :

$$g = G_1(q) = \alpha^{(p-1)q} \bmod p,$$

40 Schritt (b) ein Schritt der Berechnung der Funktion V durch die folgende Gleichung ist:

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L) = X_1^d I_1^e \dots X_L^d I_L^e \bmod p, \text{ und}$$

45 Schritt (c) ein Schritt ist zur Berechnung des akkumulierten Werts Y durch

$$Y = \sum_{i=1}^L y_i \bmod q,$$

50 und Berechnen der Funktion Γ durch die folgende Gleichung:

$$W = \Gamma(Y * g) = g^Y \bmod p.$$

67. Speichermedium nach Anspruch 59 oder 60, bei dem in dem Programm: der Parameter q ein Parameter eines Definitionsfeldes $GF(q)$ einer elliptischen Kurve $E_{a,b}(GF(q))$ ist, und, wenn ein Basispunkt einer Ordnung k auf der elliptischen Kurve durch den Parameter β repräsentiert wird und ein Parameter der elliptischen Kurve durch $a, b \in GF(q)$, diese Parameter $\{q, a, b, P, k\}$ als öffentliche Systeminformation veröffentlicht werden, und wobei:

Schritt (b) ein Schritt ist zum Erhalt von Z' durch sequentielle Berechnung von Werten $(X_i * d_i)$, erhalten durch d_i -malige Multikomponenten-Gruppenberechnungen von X_i , und Werten $(I_i * e_i)$, erhalten durch e_i -malige Multikomponenten-Gruppenberechnungen von I_i , für jedes i von 1 bis L , und
 60 Schritt (c) ein Schritt ist zur Berechnung von W durch y_L -malige Berechnungen des Parameters P auf der elliptischen Kurve unter Verwendung der empfangenen Information y_L und der öffentlichen Information P .

68. Speichermedium nach Anspruch 67, bei dem in dem Programm: die Funktion G_1 zur Berechnung des Parameters $\beta = P$ im voraus gegeben ist durch die folgende Gleichung:

$$G_1(q) = P \in E_{a,b}(GF(q)),$$

Schritt (b) ein Schritt ist der Berechnung der Funktion V durch die folgende Gleichung:

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L) = (d_1 X_1 + \dots + d_L X_L + e_1 I_1 + \dots + e_L I_L) \text{ über } E_{a,b}(GF(q)), \text{ und}$$

Schritt (c) ein Schritt ist der Berechnung der Funktion Γ durch die folgende Gleichung:

$$W = \Gamma(y_L * P) = y_L P \text{ über } E_{a,b}(GF(q)).$$

69. Speichermedium nach Anspruch 61, bei dem in dem Programm: der Parameter q ein Parameter eines Definitionsfeldes $GF(q)$ einer elliptischen Kurve $E_{a,b}(GF(q))$ ist, und, wenn ein Basispunkt einer Ordnung k auf der elliptischen Kurve durch den Parameter β repräsentiert wird und ein Parameter der elliptischen Kurve durch $a, b \in GF(q)$, diese Parameter $\{q, a, b, P, k\}$ als öffentliche Systeminformation veröffentlicht werden, und bei dem:

Schritt (b) ein Schritt ist zum Erhalten von Z' durch sequentielle Berechnung von Werten $(X_i * d_i)$, erhalten durch d_i -malige Multikomponenten-Gruppenberechnungen von X_i und Werten $(I_i * e_i)$, erhalten durch e_i -malige Multikomponenten-Gruppenberechnungen von I_i , für jedes i von 1 bis L , und

Schritt (c) ein Schritt der Berechnung von W ist durch y_L -malige Berechnungen des Parameters P auf der elliptischen Kurve unter Verwendung der empfangenen Information y_L und der öffentlichen Information P .

70. Speichermedium nach Anspruch 69, bei dem in dem Programm: die Funktion G_1 zur Berechnung des Parameters $\beta = P$ im voraus gegeben ist durch die folgende Gleichung:

$$G_1(q) = P \in E_{a,b}(GF(q)),$$

Schritt (b) ein Schritt der Berechnung der Funktion V durch die folgende Gleichung ist:

$$Z' = V((X_i * d_i), (I_i * e_i) \mid i = 1, \dots, L) = (d_1 X_1 + \dots + d_L X_L + e_1 I_1 + \dots + e_L I_L) \text{ über } E_{a,b}(GF(q)), \text{ und}$$

Schritt (c) ein Schritt der Berechnung der Funktion Γ durch die folgende Gleichung ist:

$$W = \Gamma(Y * P) = YP \text{ über } E_{a,b}(GF(q)).$$

wobei

$$Y = \sum_{i=1}^L y_i \text{ mod } k.$$

Hierzu 19 Seite(n) Zeichnungen

FIG. 1A

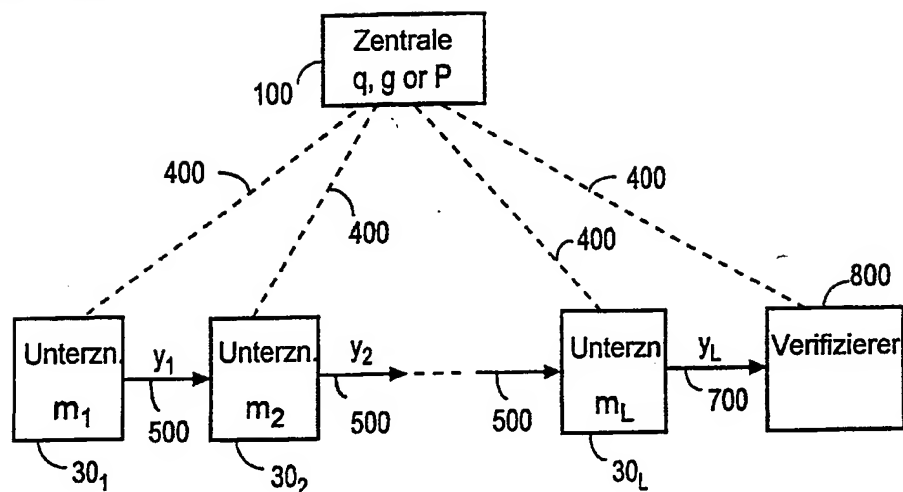


FIG. 1B

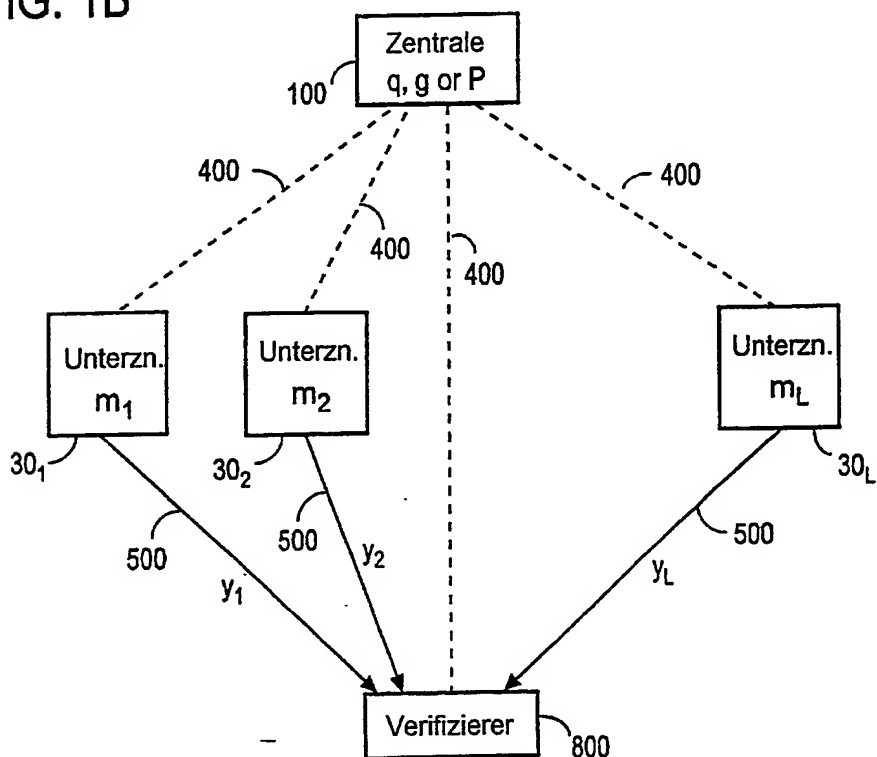


FIG. 2

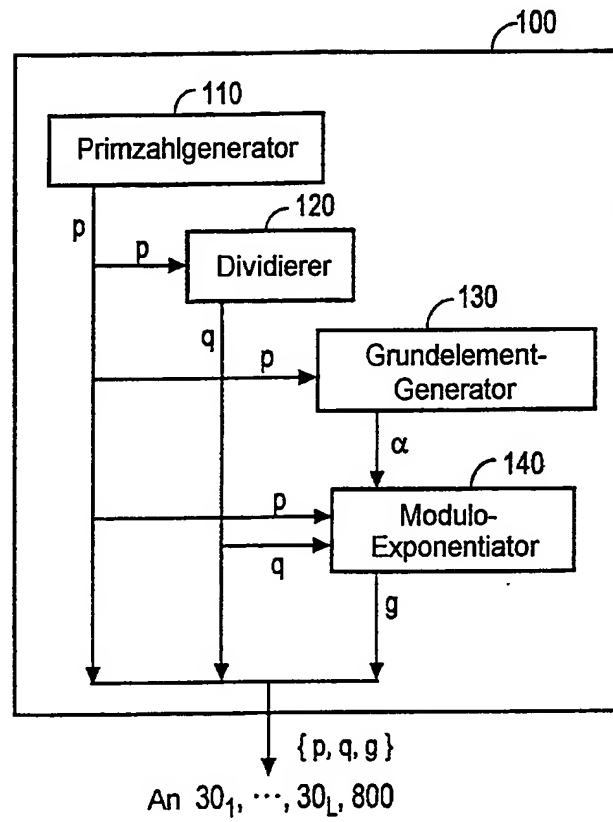


FIG. 3

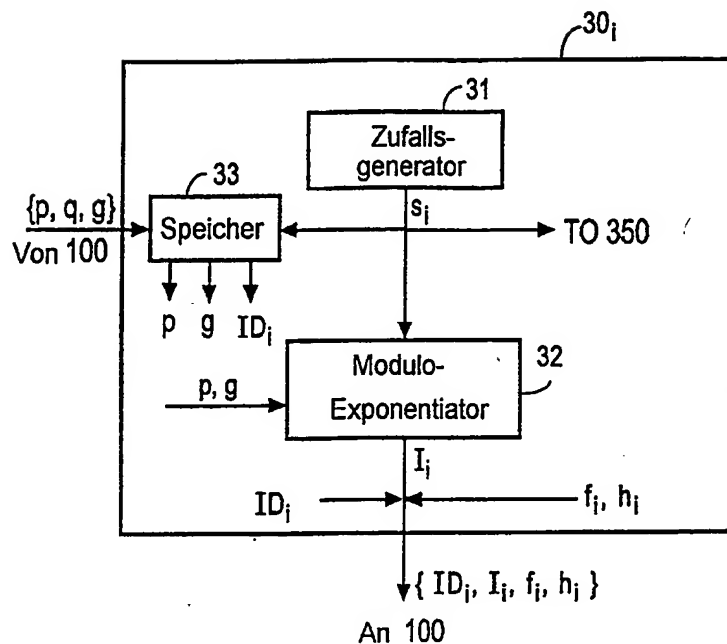


FIG. 4

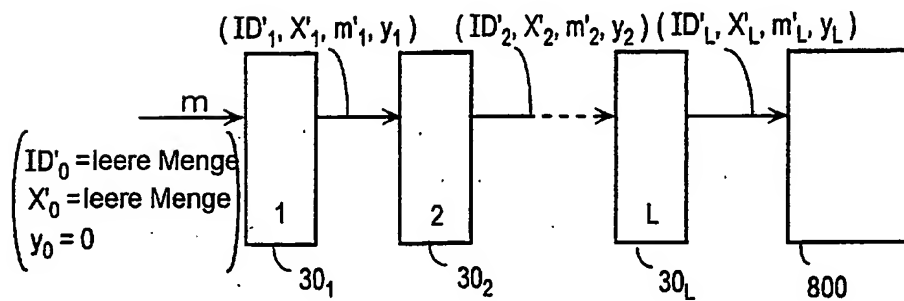


FIG. 5

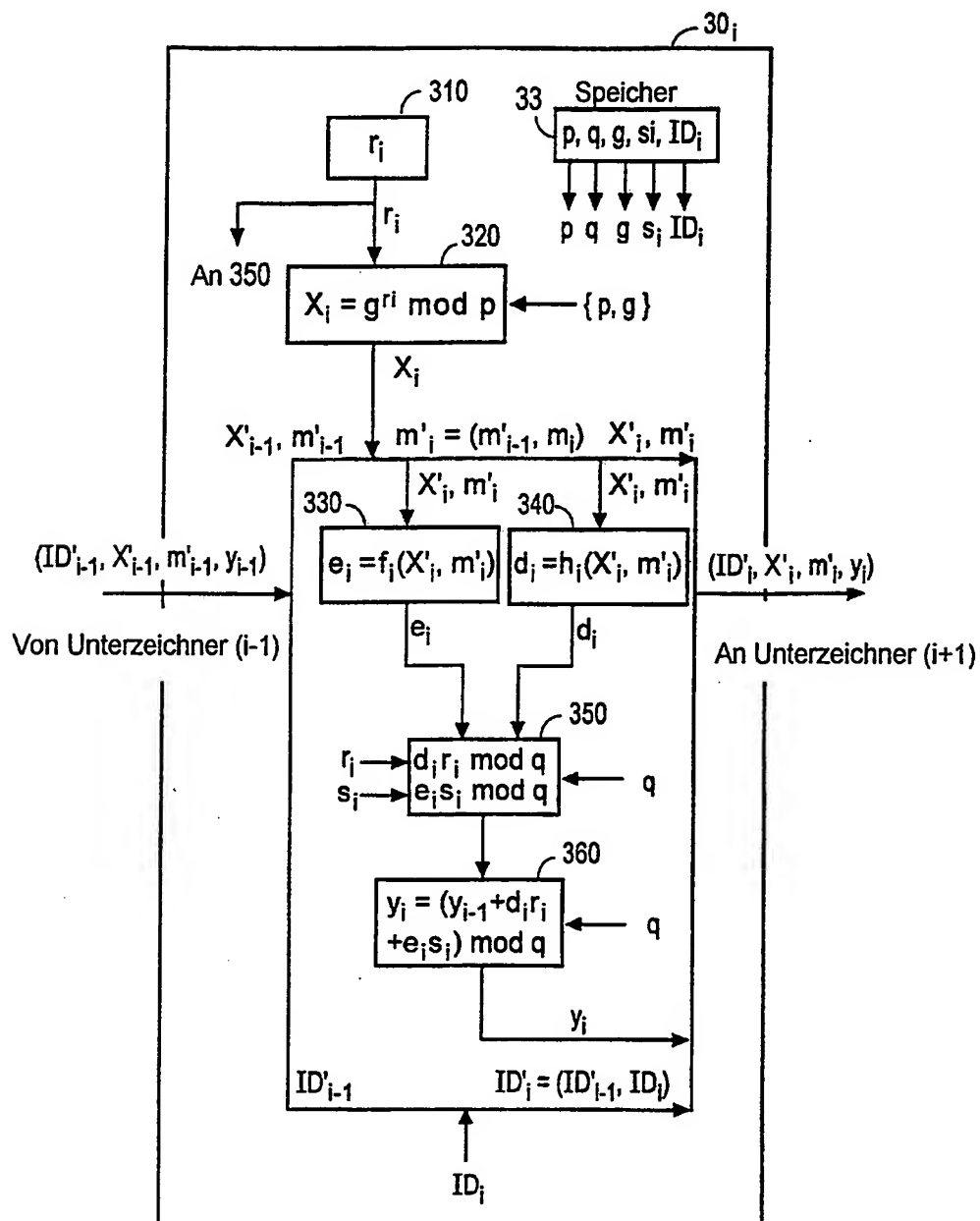


FIG. 6

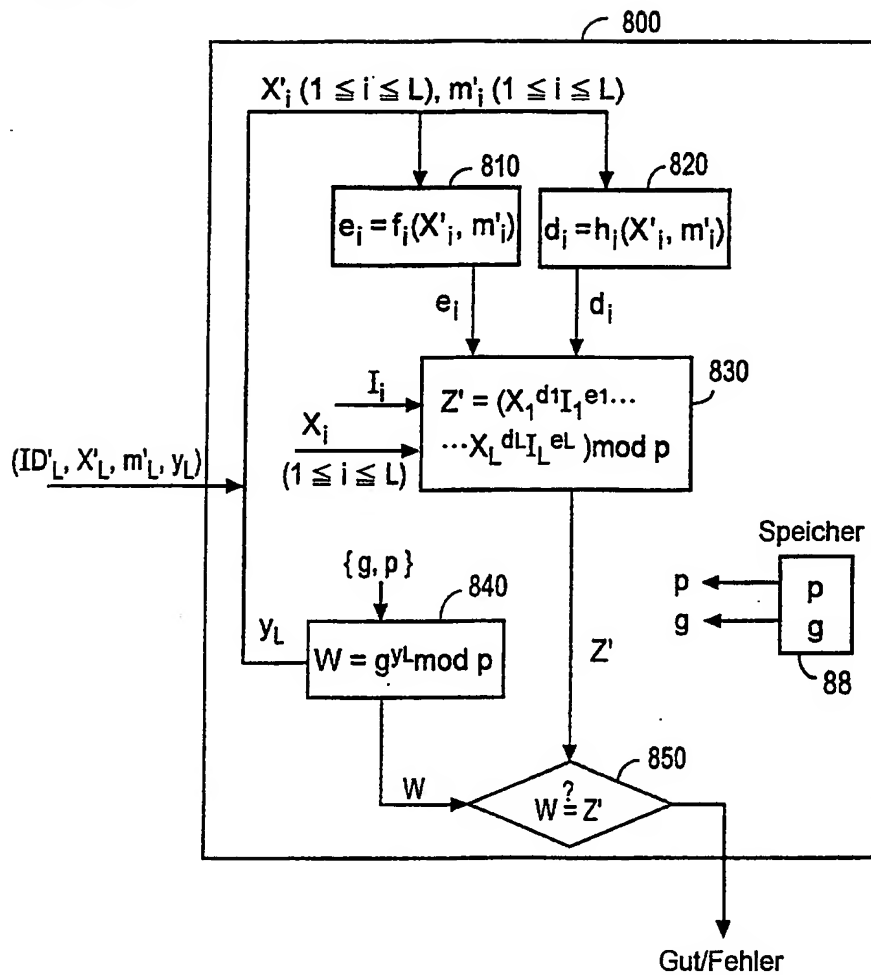


FIG. 7

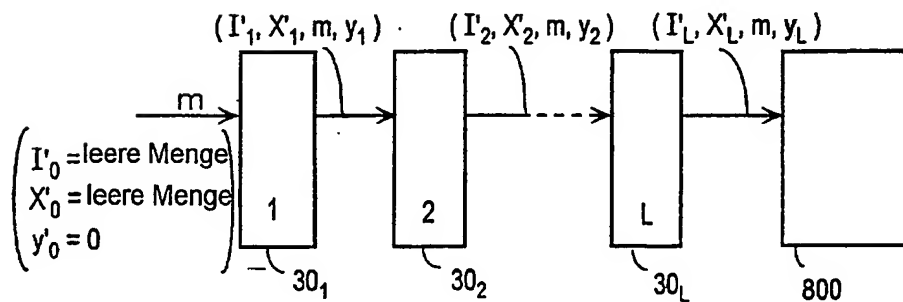


FIG. 8

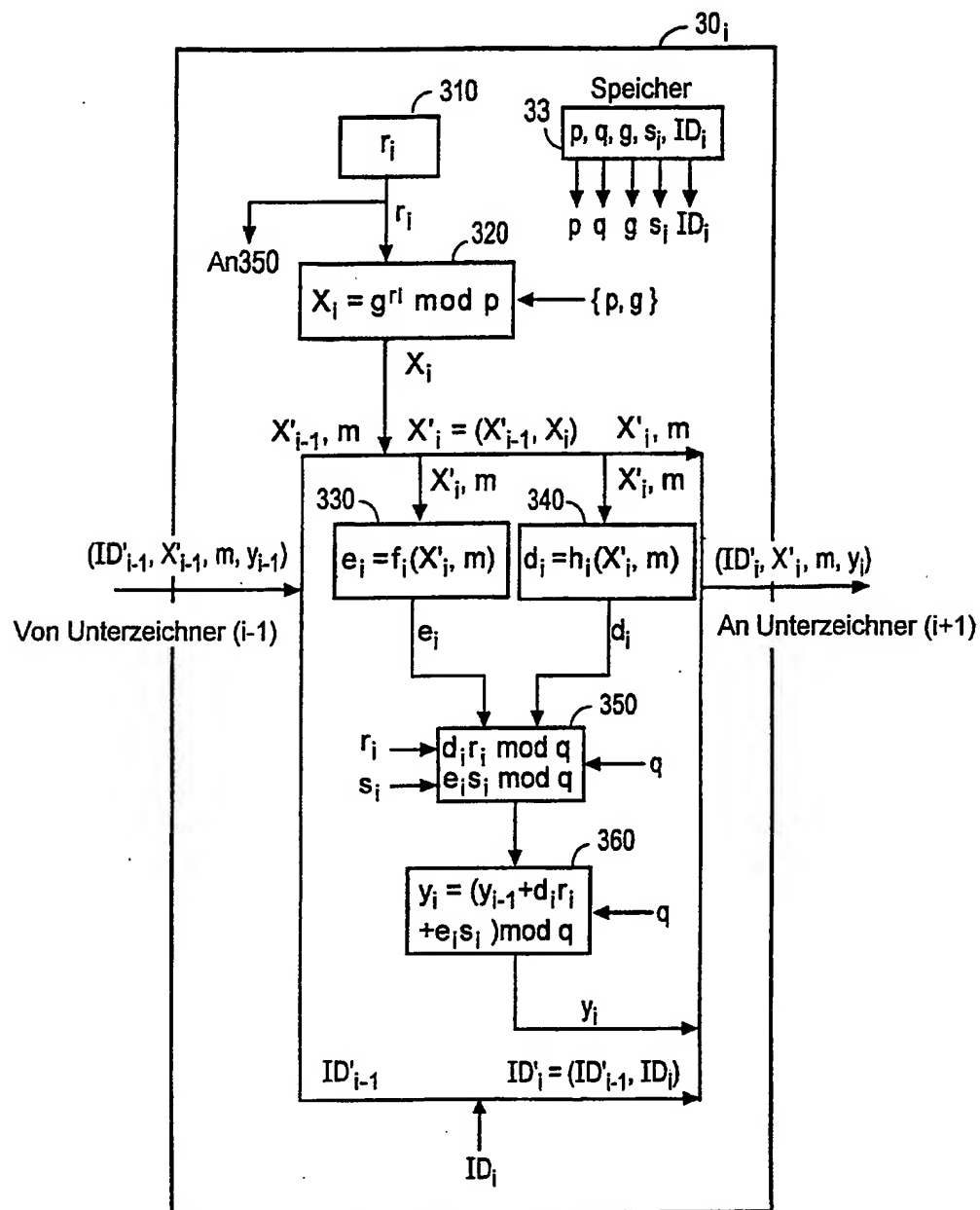


FIG. 9

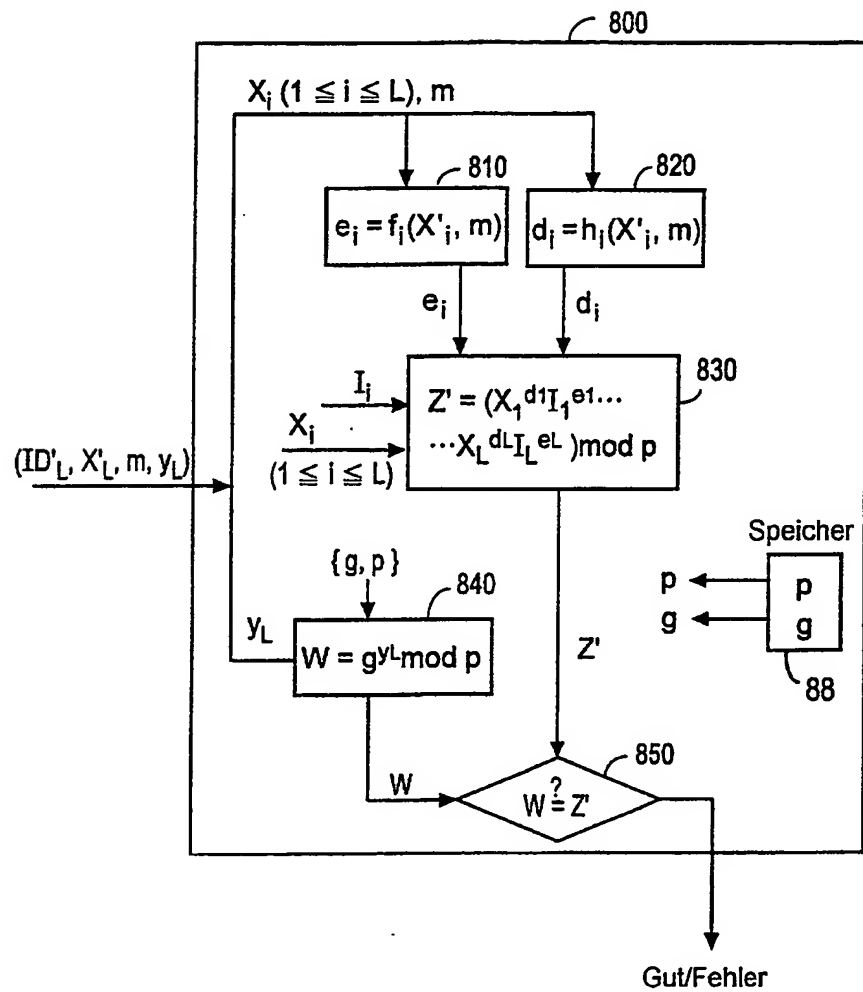


FIG. 10

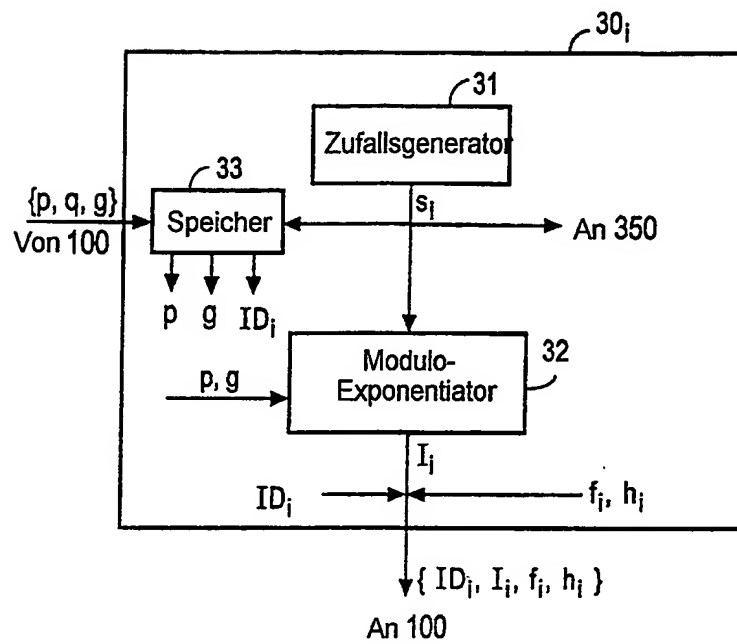


FIG. 11

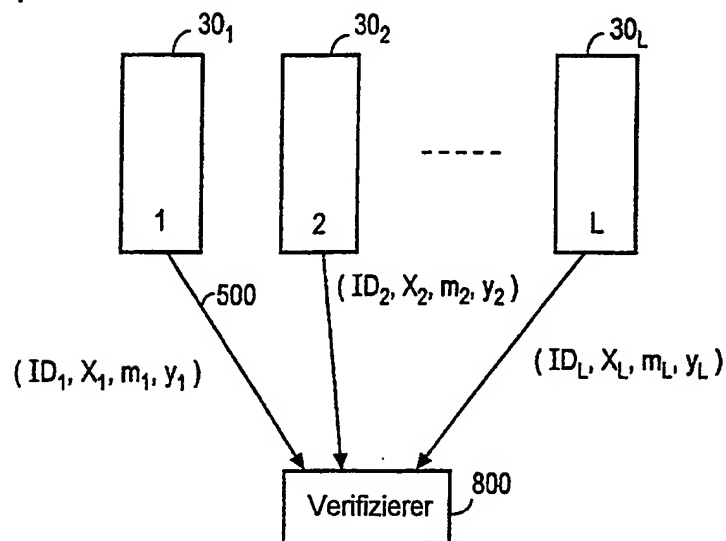


FIG. 12

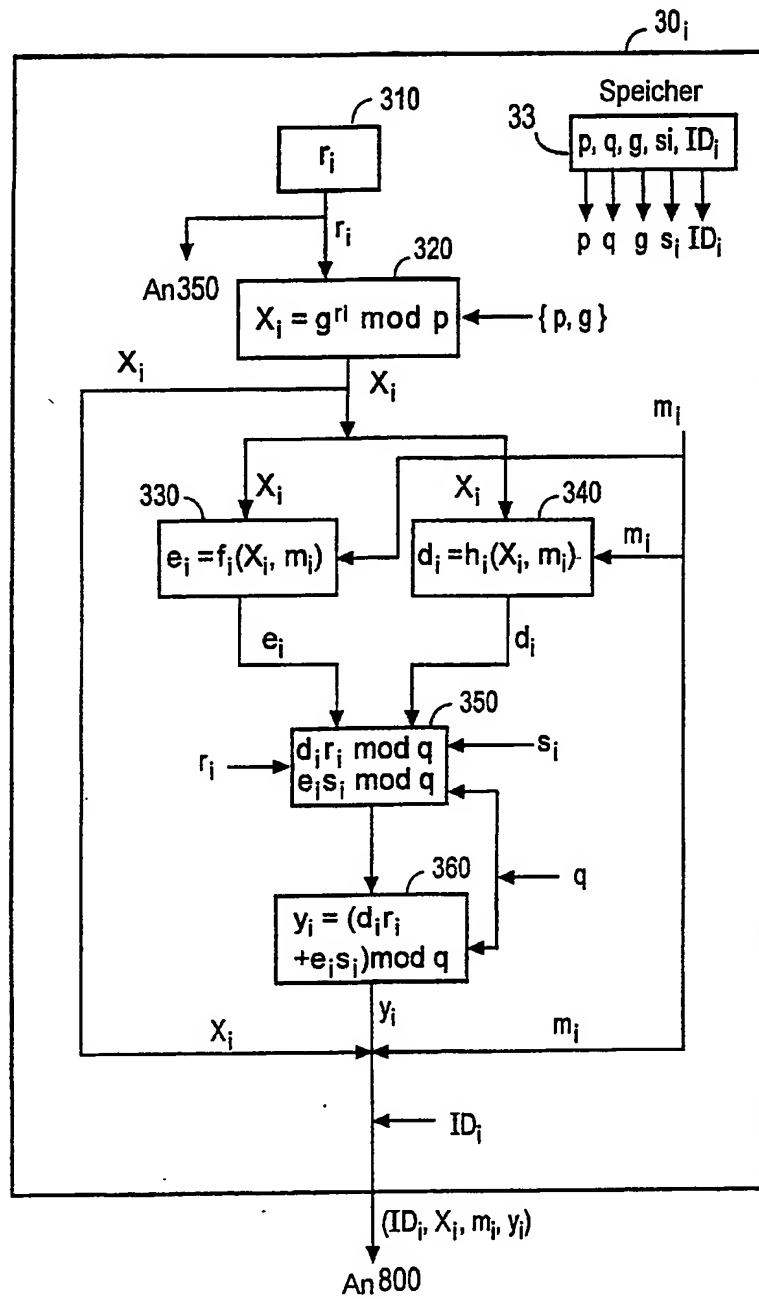


FIG. 13

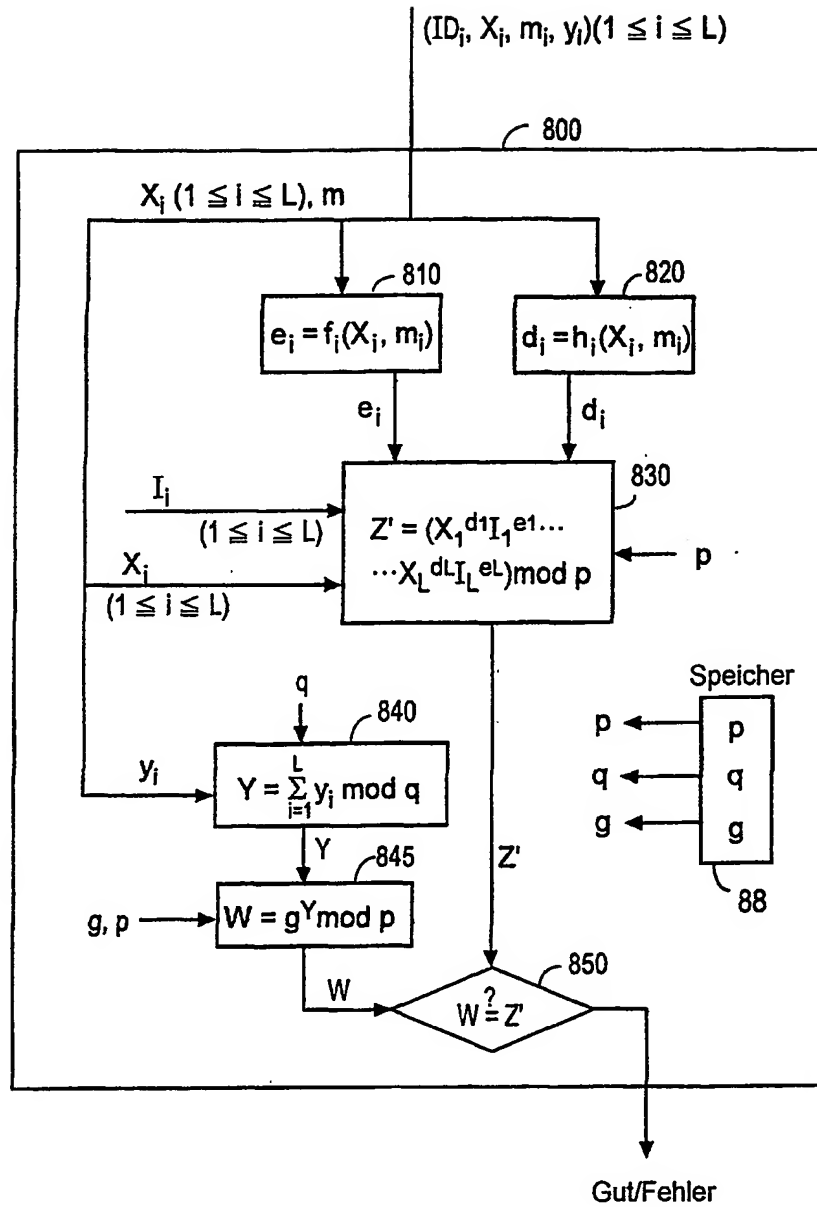


FIG. 14

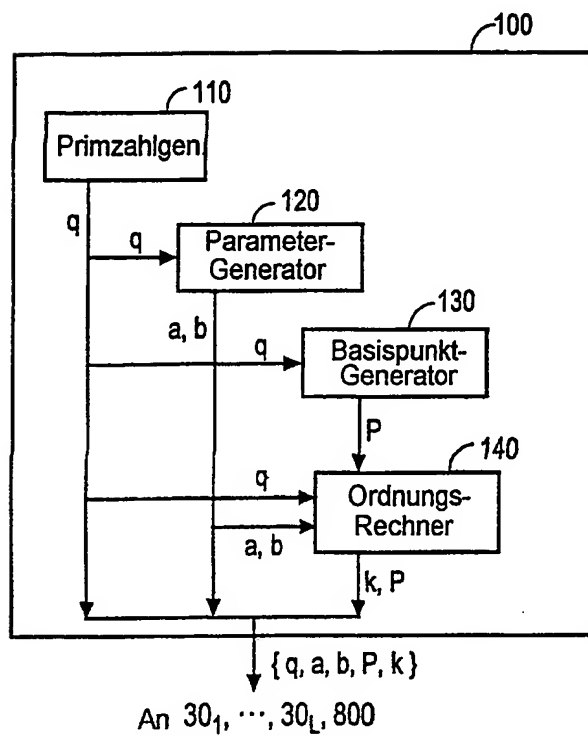


FIG. 15

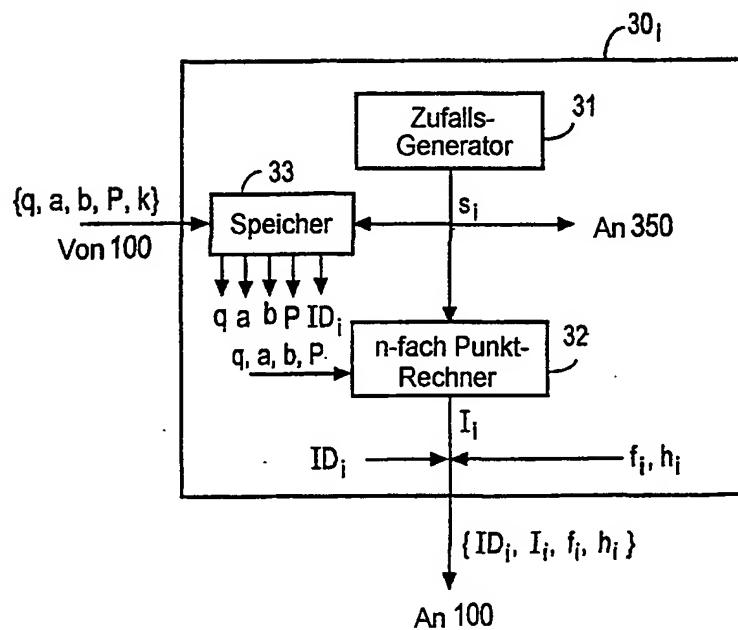


FIG. 16

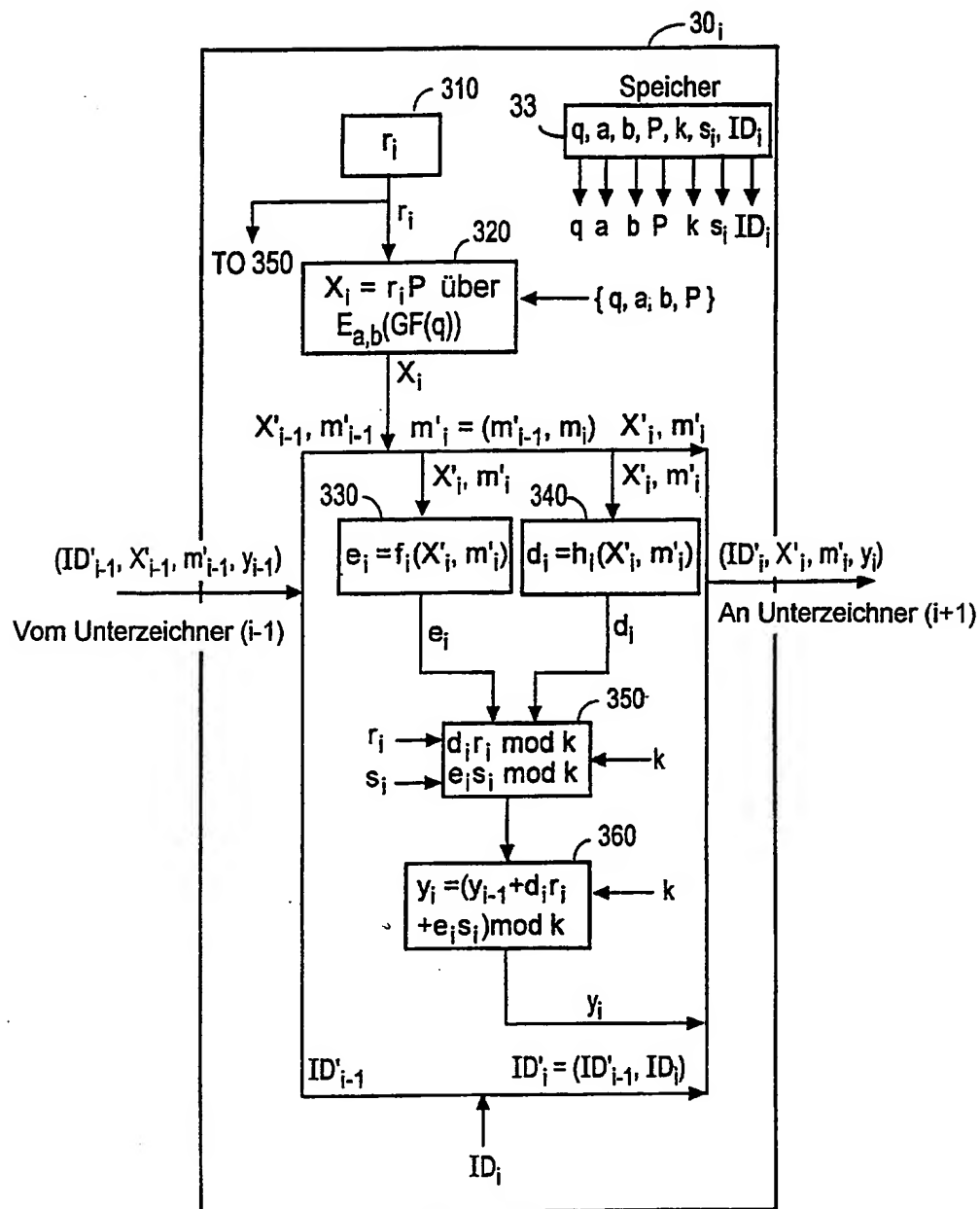


FIG. 17

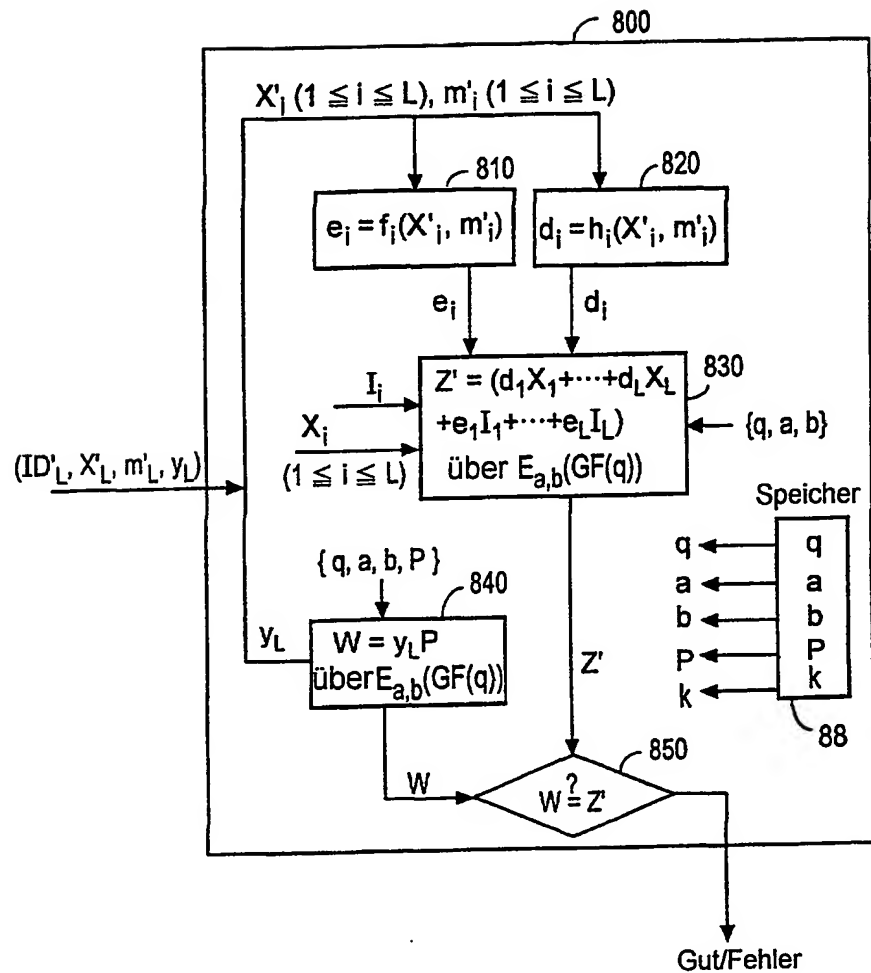


FIG. 18

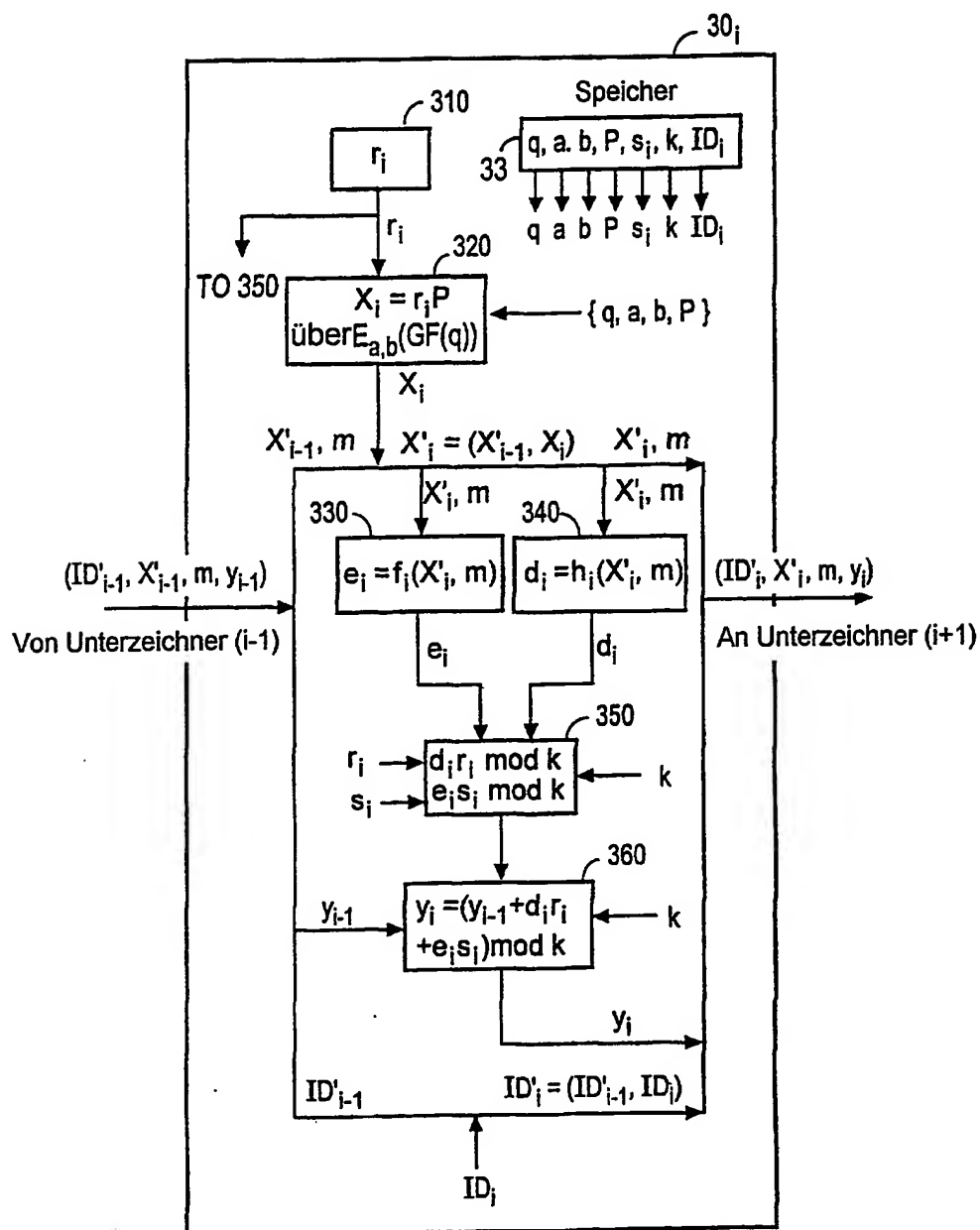


FIG. 19

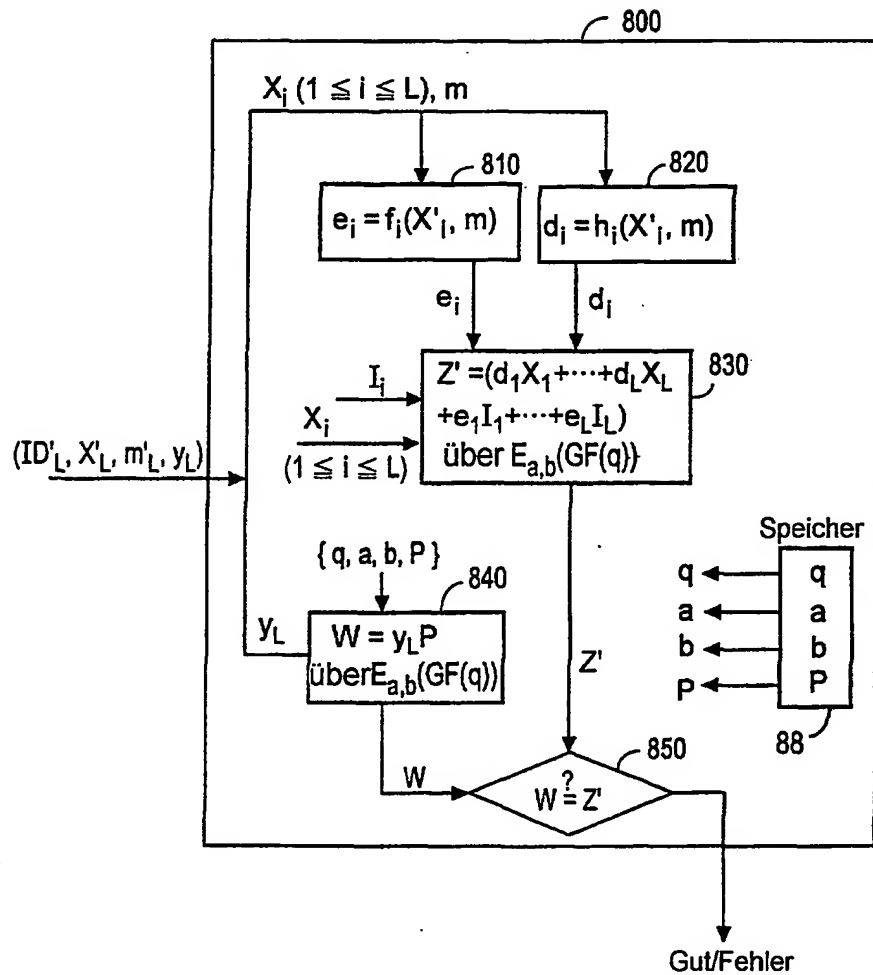


FIG. 20

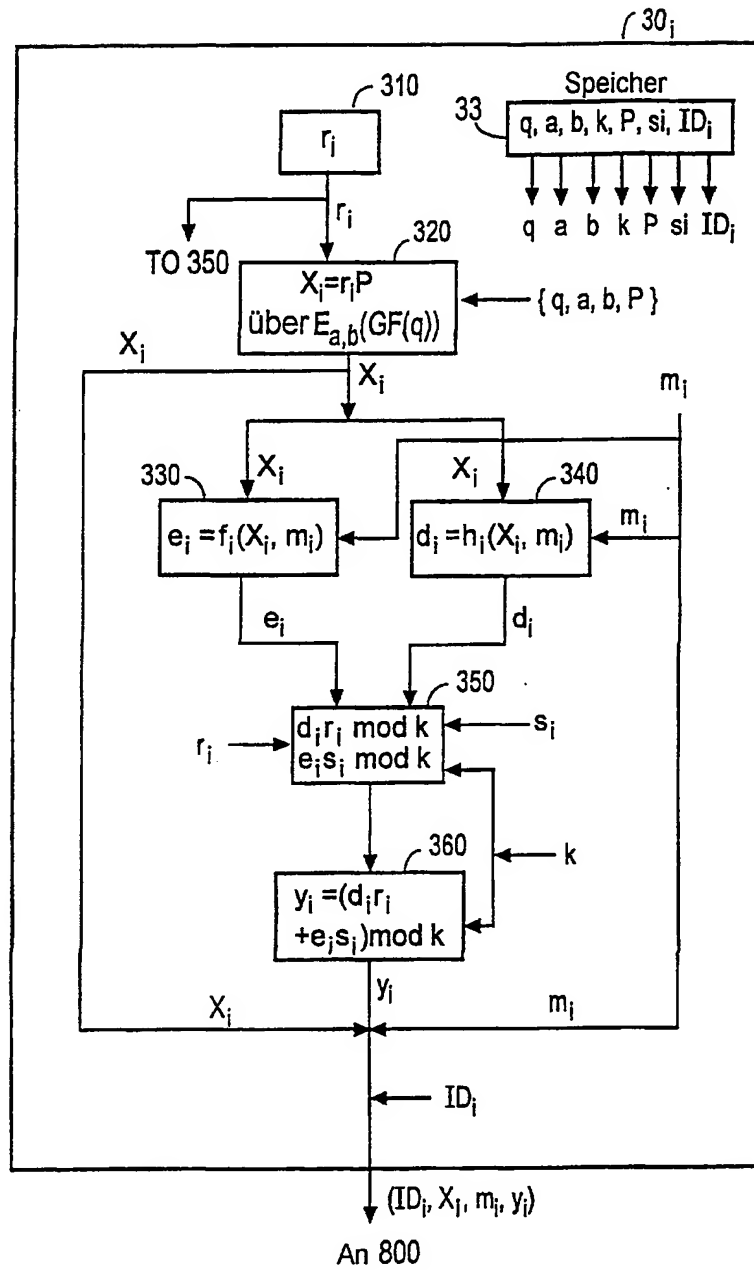


FIG. 21

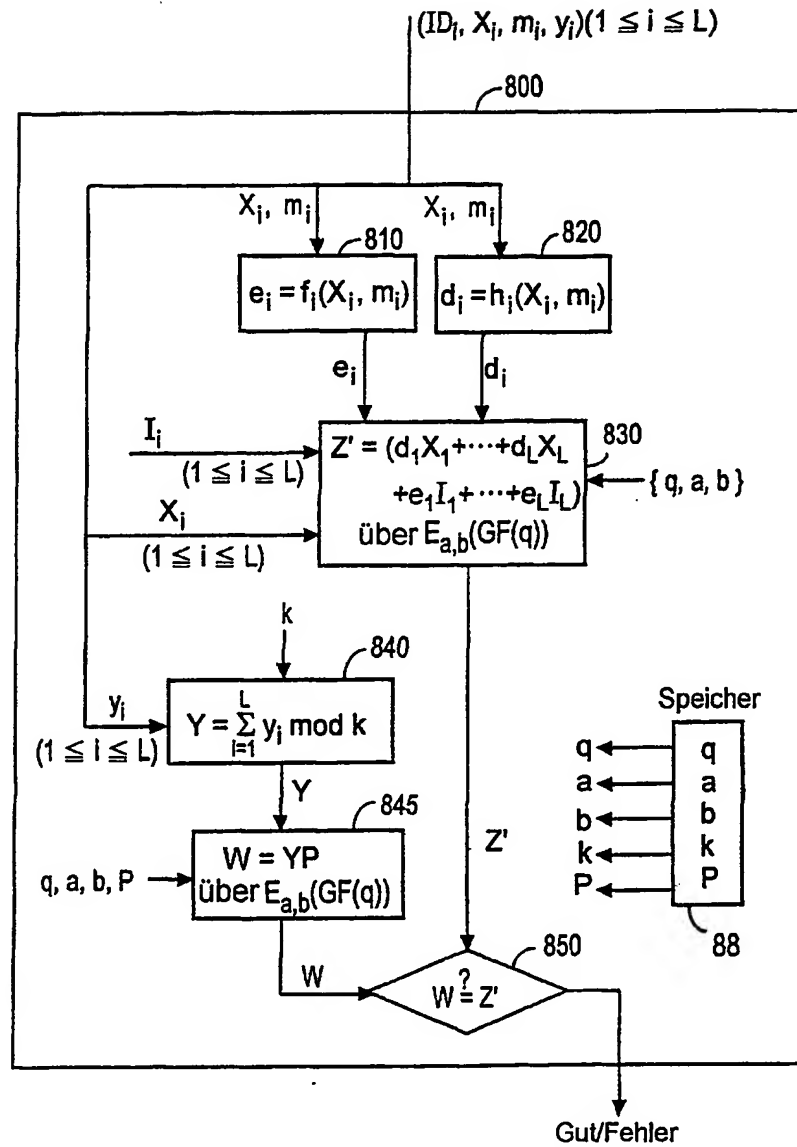


FIG. 22

	Signatur	Verifikation
RSA-Verschl. System (Stand d. Technik)	$\sigma_1 = D_1(f(m))$ for $i=1$ $\sigma_i = d_i(\sigma_{i-1})$ for $2 \leq i \leq L$	$f(m) = E_1(\dots(E_L(\sigma_L)\dots))$
Schnorr (Stand d. Technik)	$X_i = g^{r_i} \text{ mod } p$ $e_i = f_i(X_i, m)$ $y_i = (r_i + e_i s_i) \text{ mod } q$	$g^{y_i} \equiv X_i I_i^{e_i} \text{ (mod } p)$ $1 \leq i \leq L$
Ausführungs- Beispiel 2	$X'_i = X'_{i-1} g^{r_i} \text{ mod } p$ $e_i = f_i(X'_i, m)$ $d_i = h_i(X'_i, m)$ $y_i = (y_{i-1} + d_i r_i + e_i s_i) \text{ mod } q$	$g^{y_L} \equiv X_1^{d_1} \dots X_L^{d_L} I_1^{e_1} \dots I_L^{e_L} \text{ (mod } p)$
Ausführungs- Beispiel 5	$X_i = r_i P \text{ over } E_{a,b}(GF(q))$ $e_i = f_i(X'_i, m)$ $d_i = h_i(X'_i, m)$ $y_i = (y_{i-1} + d_i r_i + e_i s_i) \text{ mod } k$	$y_L P \text{ over } E_{a,b}(GF(q)) \equiv (d_1 X_1 + \dots + d_L X_L + e_1 I_1 + \dots + e_L I_L) \text{ über } E_{a,b}(GF(q))$

FIG. 23

	Signatur $ N =1024, q =160$	Verifikation	Redundanz	Komm.#	Runden#
RSA-Verschl. System (Stand d. Technik)	mod N Multiplikation $3 N /2=1536$ Mal	mod N Exponentiation L Mal	$ y $ BITS	(L-1)	1
Schnorr (Stand d. Technik)	mod p Multiplikation $3 p /2=240$ Mal mod q Multiplikation 1 Mal mod q Multiplikation 1 Mal	mod p Exponentiation 2L Mal	$L(e + y)$ BITS	2L-1	2
Ausführungs- Beispiel 2	mod p Multiplikation $3 q /2=240$ Mal mod q Multiplikation 2 Mal mod q Multiplikation 2	mod p Exponentiation L+1 Mal	$L X + y $ BITS	L-1	1
Ausführungs- Beispiel 5	mod q Multiplikation $3 q /2=240$ Mal mod k Multiplikation 2 Mal mod k Multiplikation 2 Mal	mod q Exponentiation L+1 Mal	$161L+160$ BITS	L-1	1